Reference Architectures for Smart Learning Environment

A thesis submitted during 2021 to the University of Hyderabad in partial fulfillment of the award of a Ph.D. degree in Computer Science

by

G R ANIL

SCHOOL OF COMPUTER & INFORMATION SCIENCES
UNIVERSITY OF HYDERABAD
(P.O) CENTRAL UNIVERSITY
HYDERABAD, TELANGANA-500046, INDIA

April 21, 2021

CERTIFICATE

This is to certify that the thesis entitled **"Reference Architectures for Smart Learning Environment"** submitted by **G R Anil** bearing registration number **15MCPC09** in partial fulfilment of the requirements for award of **Doctor of Philosophy** in the **School of Computer and Information Sciences** is a bonafide work carried out by him under my supervision and guidance.

This thesis is free from plagiarism and has not been submitted previously in part or in full to this or any other university or institution for award of any degree or diploma.

Further, the student has the following publications before submission of the thesis for adjudication and has produced evidence for the same in the form of acceptance letter or the reprint in the relevant area of his research:

1. "Blockchain Enabled Smart Learning Environment Framework," Advances in Decision Sciences, Image Processing, Security and Computer Vision.LAIS,vol 4. Springer.ISBN:978-3-030-24317-3 [Book chapter]
2. Formal Verification of a Block chain-enabled Smart Learning Environment Framework using Petri NETs"", *International Journal of Future Generation Communication and Networking* Vol. 13, No. 2, 2020 pp.838-853 ISSN: 2233-7857 [Index: ESCI]
3. "Personalized Dynamic Learning Plan Generator for Smart Learning Environments", *International Journal of Recent Technology and Engineering (IJRTE)* Volume-8 Issue-2, July 2019. ISSN: 2277-3878 [Index: SCOPUS]
4. "A holistic rubric for assessment of software requirements specification," 2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH), Hyderabad, 2017, pp. 1-6, ISBN:978-1-5386-1923-0. [Index: SCOPUS]

Further, the student has passed the following courses towards fulfillment of course work requirement for Ph.D.

Course Code	Course Name	Credits	Pass/Fail
CS 801	Data Structures and Algorithms	4	Pass
AI 820	Digital Image Processing	4	Pass
CS 821	Trends in Software Engineering	4	Pass
AI 852	Learning and Reasoning	4	Pass

Prof. Salman Abdul Moiz

Supervisor

SCIS, UoH

Prof. Chakravarthy Bhagvati

Dean

SCIS, UoH

DECLARATION

I, **G R Anil**, hereby declare that this thesis entitled **" Reference Architectures for Smart Learning Environment "** submitted by me under the guidance and supervision of **Prof. Salman Abdul Moiz** is a bonafide research work. I also declare that it has not been submitted previously in part or in full to this university or any other university or institution for the award of any degree or diploma.

Signature of the Student

Name: **G R Anil**

Reg. No.: **15MCPC09**

This thesis is wholeheartedly dedicated to our beloved parents, who have been our source of inspiration and gave me strength to resurrect my shortcomings and who also continually provide their moral, spiritual, emotional and financial support.

To my beloved family, friends and well wishers, whose faith and confidence kept me going to carry my research successfully. And of course, it has come this long way with the blessings of almighty God.

ACKNOWLEDGEMENTS

I would like to acknowledge my indebtedness and render my warmest thanks to my supervisor Prof Salman Abdul Moiz who made this possible. His friendly guidance, motivation, and expert advice have been invaluable in all stages of my research work. He made me a complete person. He got me to learn about society, ethics and professionalism. His patience, politeness, calmness, balancing profession and family have shown a great impact on my personality and greatly benefitted from these qualities. His approach towards the work, preparedness to take any responsibilities and his multi tasking abilities always amazes and motivates me to drive my professional life. I am truly fortunate to spend quality time with you to discuss various facets of research and also about the philosophy of life, which further helped me to enhance my knowledge and also equipped me with a great amount of required skills. He parented, monitored, cared and valued me throughout my journey as a research scholar. I am blessed to have him as my supervisor.

I would like to extend my deepest gratitude to Prof. Atul Negi, who is really my well wisher since my masters. His encouragement in the E-Learning domain is invaluable. He encouraged me to explore the Learning Management Systems and gave me experience of administering in real time as well. He introduced me to Moodle, through his provided motivation, I was able to participate in Moodle moot, where I got to know many insights of the trends in global e-Learning. Besides, he is a visionary, very informative and encouraged me throughout my phd journey. I am always thankful to him.

I would like to thank one of the other RAC members Dr.Swaroopa Rani, for being parently guiding me and for her valuable suggestions. Her kindness and helping the needy inspired me alot and I was blessed to have such a kind person in my RAC. I am fortunate to have such a decent Research Advisor Committee. This thesis is greatly benefited from comments and suggestions offered by the committee. I am grateful for their valuable contribution.

I am deeply indebted to all my teachers, who have greatly shaped my personality of what I am today and their ethically-embedded teachings have influenced my life to a great degree to put into practice all my life, which also reflects in my research as well. I would like to take this opportunity to thank all teachers and mentors of my life, especially to Smt Thippamma, Smt. Karanam Padmavathamma, Sri. B.K.S. Pranesa Rao, Sri Venugopala murthy, Dr. Kolikapudi Srinivasa Rao and to name a few others.

I am thankful forever to Ravindra Babu Ravula, Naveen Desireddy and Smt. Potturi Prameela who inspired and supported me in my hardships.

I thank my friends Abdul Basit, Devidas, Shivakrishna, Alammari, Rohit Bond, Kakarla Rajesh, Anand, Anil Kumar, Bheekya, Gaurav, Rakesh, Edukondalu, Abhay Pradhan, Kashi, Thirupathi and all my co research scholars for their contribution towards my research, which helped me to set up a comfortable research environment all the time. I also thank my friends Dadapeer, Govardhan Naidu, D.V.S Raviteja, Nagesh, Sudheer, Sridhar, Hareesh, Praveen, Swamy, Prathyusha, Venkatesh Terli, Bhavana and each one supported me and being my all time well wishers.

I am truly blessed to have friends Naresh Nanduri, Prahakar Podapati, Arjun Guguloth, Hita Gangineni, Ravi Kiran G Ch, Madhu Pasula, G T Naik, Satish Doppalapudi, and Mounika Venna who are always with me and believed and supported me. Their unconditional love and rock solid affection always strengthened me. I miss my dearest Naresh Nanduri forever, who always wished for my best. I am sure he must be smiling from the clouds on this day of PhD submission.

I extend my deepest salutation to my whole family members Dr. Lakshmi Devi, my brother-in -law Prof. Tharakeswar, my brother Manoranjan for providing the needful facilities and emotional assistance. I am grateful to my sister, whom I consider as a mother, and to my brother-in-law for showing the path towards Research. And also for their constant support and consistent encouragement helped me to stay strong at every stage of life during research..

I wish to express that I am indebted and owe so much to my parents Smt.. Gandhudi Sarada Devi and Sri. Gandhudi Ramanjineyulu, where my depth of expressing acknowledgement will always be short, when to compare their contribution of parenting. The unconditional sacrifices and determined motivation is what I have imbibed at every stage of my life, which also has enforced me to become a responsible citizen in the first place. However, with an immense pride this doctoral dissertation is dedicated to my parents for providing me the strong roots to stand alone and sustain in such a long journey of life.

Finally, I bow The Almighty for being source of unknown power and strength in my life.

G R Anil

ABSTRACT

Smart learning is concerned with context-aware ubiquitous learning. Contexts include the interactions between learners and the learning environments. Smart learning environments comprise technology-enhanced learning environments to fulfill the need to provide the right content at the right time. Smart learning provides holistic learning to students with a significant paradigm shift by using modern-day technologies. The Learning Management systems are primarily build using traditional standard frameworks like IEEE Learning Technology Systems Architecture (LTSA).

In the online learning domain, the IEEE standard of Learning Technology Systems Architecture is a widely known framework proposed in the year 2003. It provides a generic Software architecture for the Learning Management systems. In the same year, IMS Global Learning Consortium has proposed IMS abstract framework. MIT has proposed OKI(Open Knowledge Initiative). Later, the OKI was dropped from MIT's archives due to its impediments. A universally accepted framework stemmed from the Joint Information Systems Council (JISC) from the UK, which has the online learning research community's approval. However, these are not regarded as smart learning frameworks. The smartness level is a measure to assess the smartness of the smart learning environments. There are certain Standardization challenges, which are associated with the six Smartness levels. Each level is associated with certain Standardization challenges. The IEEE LTSA is at the Pre-Smart level(i.e., Smartness Level 0). The Smart Learning Environments requirements are also specified by the ISO/IEC ITLET (Information Technology for Learning, Education, and Training) subcommittee (here on wards SC36).

In the Smart Learning Environments literature, there are few normative models proposed, like Hwang, Spector, Liu, Koper, etc. These are conceptual, thus can not be Reference Architectures. A Reference architecture provides a methodology or set of practices and templates based on the generalization of a set of successful solutions. Reference architectures provide guidance to apply specific patterns or techniques to solve particular classes of problems.

In this way, it serves as a "reference" for the specific architectures that companies will implement to solve their problems. It is never intended that a reference architecture implementation as-is but rather used either as a point of comparison or as a starting point for individual companies' architectural efforts.

In light of the above limitations and recommendations, there is a requirement for new-age frameworks for the Smart Learning Environments which address the Smartness standardization challenges. This research gap addressed in this Research work is by extending to the IEEE standard for LTSA towards Standard for Smart Learning Environments. The Smartness challenges in the first two levels like Security, Interoperability, and more significantly Personalization are addressed in this Research work. Each challenge is addressed with a Reference Architecture in Conformance to the IEEE LTSA and expressed in Architecture Description Languages(ADL). Architecture diagrams are generated with the Architecture Description Languages and Validated with that ADL's specific tool. The Architecture Analysis and Description Language(AADL) is used for expressing architectures, and the OSATE tool for the AADL is used for validation. Scenario-based verification is performed on the proposed Reference Architectures.

The first challenge, "Security," is addressed through Blockchain-enabled Smart Learning Environment Reference Architecture. The ISO/IEC ITLET SC36 has recommended the usage of Blockchain. The Smart Learning Environments literature highlighted the lack of a blockchain framework in this domain. Thus a Reference Architecture is proposed with distributed database and Validated with Architecture Description Language. The most vulnerable learning activity -Examination Scenario is considered in this contribution. According to the proposed Reference Architecture, a workflow for the online examination Scenario is designed and formally verified with Petri Nets.

The second challenge Interoperability is addressed with a Reference Architecture by designing a middleware consisting of multiple interoperability modules to the IEEE LTSA. IEEE Layered structure for LTSA is considered, and an additional layer is introduced. The proposed Reference Architecture is expressed with AADL and validated with OSATE. An activity-based Inteorperablity module selection Scenario is considered for the proposed Reference Architecture. It is verified with Colored Petri Nets.

The most significant challenge "Personalization" is addressed with a design of Intelligent Coach to the IEEE LTSA. The smartness levels from three to six involve various artificial intelligence activities. To facilitate this, an additional component named "Intelligent component" is added to the IEEE LTSA. The pipe-Filter reference model is adopted for this purpose. Proposed Reference Architecture is expressed using AADL and validated. For Scenario-based verification, "Personalized dynamic learning plan generation" is considered and realized formally with Petri nets.

The proposed Reference architectures help the organizations that develop Smart Learning Management Systems. As they are expressed in Architecture Description Languages, the designers can proceed with pre-analysis with their specific configurations. These formally verified scenarios help as a reference framework for the implementation of that particular tasks. There are open calls from the e-Learning standards bodies for contributions for the upcoming standards. This work of addressing Smartness challenges with Reference architectures can contribute to the new age standards.

Contents

List of Figures

List of Tables

ABBREVIATIONS

SLE	**S**mart **L**earning **E**nvironment
IEEE	Institute of Electrical and Electronics Engineers
LTSA	Learning Technology System Architecture
LTSC	Learning Technology Standards Committee
ITLET	**I**nformation **T**echnology for **L**earning, **E**ducation and **T**raining
SC36	**S**ub **C**ommittee 36
IASLE	**I**nternational **A**ssociation of **S**mart **L**earning **E**nvironments
ISO	**I**nternational **S**tandards **O**rganization
MIT	**M**assachusetts **I**nstitute of **T**echnology
OKI	**O**pen **K**nowledge **I**nitiative
JISC	**J**oint **I**nformation **S**ystems **C**ouncil
ADL	**A**dvanced **D**istributed **L**earning
TLA	**T**otal **L**earning **A**rchitecture
SCORM	**S**harable **C**ontent **O**bject **R**eference Model
API	**A**pplication **P**rogram **I**nterface
xAPI	e**X**perience **API**
AICC	**A**viation **I**ndustry **C**BT (Computer-Based Training) **C**ommittee
LTI	**L**earning **T**ools **I**nteroperability
CMI	**C**omputer **M**anaged **I**nstruction
PS agents	**P**rivacy and **S**ecurity agents
ADL	**A**rchitectural **D**escription **L**anguages
SPN	**S**tochastic **P**etri **N**ets
AADL	**A**rchitecture **A**nalysis and **D**escription **L**anguage

JSON	**J**avaScript Object Notation
DLP	**D**ynamic **L**earning **P**lan
DLPS	**D**ynamic **L**earning **P**lan **S**elector
DLPG	**D**ynamic **L**earning **P**lan **G**enerator

Chapter 1

Introduction

The technology-enhanced learning environments consist of Learning Management Systems that facilitate virtual learning activities. These environments are built on high-level system designs containing various components and their interactions. The standard reference frameworks provide inputs for designing a range of systems for online learning. IEEE LTSA, IMS framework, JISC, and OKI framework are some of the standard frameworks proposed by prestigious bodies like IEEE LTSC(Learning Technology Standards Committee), MIT, etc.

In the era of Smart learning, the requirements are evolving continuously due to the learning entities' challenges. The analytical capability, ubiquitous learning assistance, achieving higher learning success rates must be highly addressed with the latest frameworks using sophisticated technologies.

The researches of the Smart Learning Environments [27] [29] [33] [1] highlights the necessity of new-age frameworks. The widely accepted standard like IEEE LTSA is withdrawn [39], and it is yet to be revised. The standard development process involves contributions from the volunteers in the domain [39]. There are open calls from the LTSC for standards development, with this research attempts to address some of the requirements to contribute to the new age standards.

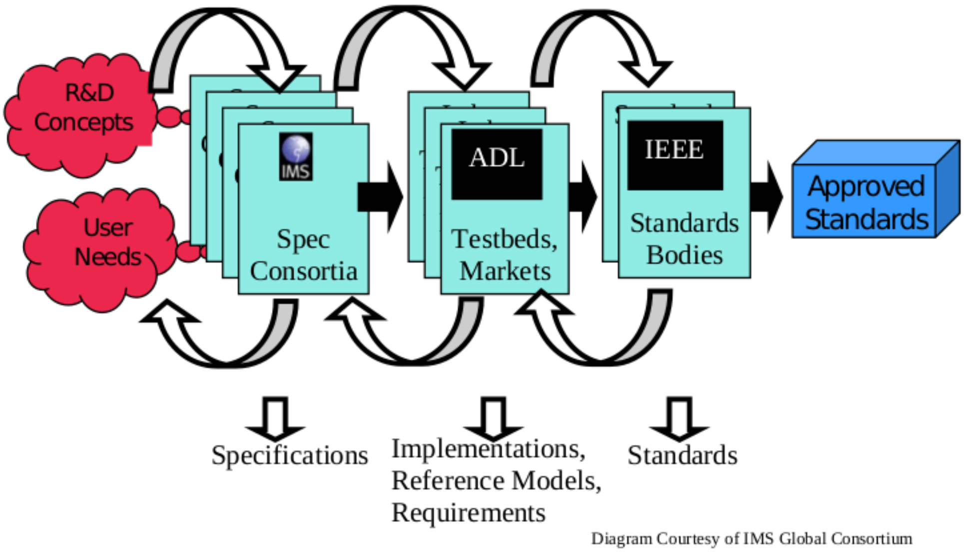


FIGURE 1.1: E-learning standards development process

1.1 Context and Motivation

Standards can be defined as "documented agreements containing technical specifications or other precise criteria to be used consistently as rules, guidelines, or definitions of characteristics, to ensure that materials, products, processes, and services are fit for their purpose" [12].

The E-Learning systems architecture standards provide high-level system designs to the developers with a proper decomposition of the learning systems into components and interactions between the components to address the E-Learning requirements.

This decomposition into components increases the independent development of the components. The standards development process is depicted in Figure 1.1, which is prescribed by IMS Global Consortium[16].

The pioneering survey works in the Smart learning environments can be classified into two models 1. Cognitive Smart learning model 2. Smartness level model [27] . These two models provide the Standardization challenges to be addressed in this area. The Standards research community attempts to address these standardization challenges with reference frameworks, algorithms for specific problems/scenarios, instructional designs, implementations of certain modules, etc. The Standard bodies evaluate these research outputs. The pooled analysis of these reference models becomes standards on approval of the standard bodies.

It is mentioned in the IEEE standard for Learning Technology Systems Architectures 1484.1:2003, [39](here on IEEE LTSA) “The IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product“ which creates an opportunity for the Standard research community to be part of the Standards development process.

The IEEE LTSA is a benchmark standard for learning systems. It was proposed in the year 2003 by the IEEE Learning Technology Standards Committee(here on LTSC). This provided a high-level layered Reference architecture with abstract components and interactions. As per the IEEE LTSA standard, any document more than five years without revision or reaffirmation is reasonable to conclude that it does not totally reflect the current state of the art. However, this standard was withdrawn in 2009. As rightly said in the standard document, it does not reflect the current requirements derived from the various educational theories and models such as Tore Hoel [27], Hwang[29], Uskov[62], Koper [33] etc.

The withdrawal of the IEEE LTSA standard, and the latest Standardization challenges motivated the research on Reference Architectures for the Smart Learning Environments.

The pioneering survey works in the Smart learning environments can be classified into two models 1. Cognitive Smart learning model 2. Smartness level model [27] . These two models provide the Standardization challenges to be addressed in this area. The Standards research community attempts to address these standardization challenges with reference frameworks, algorithms for specific problems/scenarios, instructional designs, implementations of certain modules, etc. The Standard bodies evaluate these research outputs. The pooled analysis of these reference models becomes standards on approval of the standard bodies.

There are attempts to extend the IEEE LTSA with various technical methods like cloud-based, service-oriented, etc. The current existing works are not presented or expressed in Architecture Description Languages, and it is one of the limitations for this work to be a

Reference Architecture. The Smart Learning Environments requirements are also specified by the ISO/IEC ITLET (Information Technology for Learning, Education, and Training) subcommittee (here onwards SC36). The Smartness levels proposed by Uskov[62] in 2015 gave a new dimension in assessing the Smartness of a Learning environment. It is a six-level Smartness concept with Standardization and technologies to be adopted at each level. The research by Tore Hoel and Mason[27] determined that the IEEE LTSA is at the Pre-Smart level. Thus it highlights the gap to extend the IEEE LTSA standard to be a Smart Learning Environment framework by addressing the Standardization challenges. The research work in Tore Hoel [27] also highlighted that the ISO/IEC SC36 recommendation for Blockchain in education is yet to be addressed in Reference Architectures of E-Learning Standards, which gave a research insight for us to proceed with the first step. The smartness level 1, namely "Adapt" in [62] has the Standardization challenges as " Security," "Privacy," "Data Interoperability," "Data Governance."

We have considered the most significant contemporary Smart learning environment requirements like Blockchain, Interoperability issues, and Intelligent systems for Smart Learning Environments. Reference architectures for the above requirements are proposed and validated in our research. The scenarios for each requirement are also designed and verified formally with Petri nets.

1.2 Research Study Objectives

1.2.1 Research Problem

This research aims to design Reference Architectures for Learning technology systems, which address some of the standardization challenges and contemporary technical requirements for the Smart Learning Environments. The IEEE LTSA standard framework, which is withdrawn for its limitations, is adopted as a reference model and extended to design new Reference Architectures. The Scenarios/views are designed on each architecture for a specific challenge and verify formally. Holistically, this research work aims to take the Pre-Smart IEEE LTSA towards the Smart Learning Environment standard.

The Research problem:

- To design a Reference architecture for the smart learning environments by extending the IEEE Learning Technology Systems Architecture standard.
- To address the technical challenges being mentioned in the Smartness levels in the Smart Learning Environments and ISO/IEC ITLET SC36 with architectural views
- To formally present the proposed Systems architecture views with the petrinets and to perform verification of behavioural properties and simulation.

1.2.2 Research Objectives

There are attempts to extend the IEEE LTSA with various technical methods. Their proposed models are not validated to be considered as Reference Architectures for the Standards developments. There are conceptual frameworks proposed with the decomposed requirements which are not the Reference architectures as Hwang[29], Koper[33],Spector[60],Zhu[68], Liu [37].

1. To design a Blockchain-enabled Reference Architecture with conformance to the IEEE standard for LTSA
2. To construct a Scenario for the most vulnerable activity Examination on Blockchain-enabled Smart Learning Environment proposed in the first step
3. To design a Reference Architecture with Interoperable Engine as an additional layer to IEEE LTSA that is capable of addressing standardization challenge of activity-based Interoperability standard selection
4. To construct a scenario that performs activity-based interoperability module selection
5. To design a Reference Architecture with Intelligent component to address the Personalization standardization challenge
6. To design a scenario for personalized assistance with the Personalized dynamic learning plan Generator

1.3 Research Scope and Contributions

This section enumerates the Scope of our Research and Contributions made to achieve the Research objectives

- To achieve the first objective, the IEEE LTSA is configured with distributed data storage /with an additional local node configuration. The IEEE LTSA has a centralized data storage system, which does not support a Blockchain mechanism. The integration of local nodes to the IEEE LTSA is done to realize the distributed nature of Blockchain. The proposed architecture is described and formally validated with an Architecture Description Language.

- A Scenario for Online examination in Blockchain-enabled Smart Learning Environment Reference Architecture is designed to achieve the Second objective. The designed scenario, including various possible states of the system, are represented and are verified formally with Petri nets. The formal verification confirmed the correctness of the proposed scenario as safe, Live, and Dead-lock free.

- To achieve the third objective, an additional layer for Interoperable Engine is added to IEEE LTSA. This additional layer is placed between the delivery and Learner Entity(Presentation layer). This layer is prescribed with an Interoperable engine that enables the activity-based interoperability module selection and enhances the system's modifiability.

- To achieve the fourth objective, A Scenario for the activity-based interoperability module selection is proposed. This Scenario is designed according to the Reference architecture proposed in the third objective. This Scenario is realized with the help of Colored Petri nets and verified formally. The formal verification confirmed the correctness of the scenario design as safe, Live, and Dead-lock free.

- To achieve the fifth objective, an additional module is added as a component in addition to the "Coach "process called "Intelligent Component." This component is a generic structure for personalized environments. The " Intelligent component has modules loaded with the learning models. Component-based software development can be adopted for the implementation of such systems.

- To achieve the sixth objective, a personalized assistant activity is considered. A Scenario for Personalized Dynamic Learning Plan Generator is described. It is realized with the Petri nets and verified formally. The formal verification confirmed the correctness of the scenario design as safe, Live, and Dead-lock free.

1.4 Outline of the Thesis

This thesis is organized as follows:

- Chapter 2, provides the literature review on E-Learning Standards, Smart Learning bodies, Reference Architectures for Learning Technology, Blockchain in Education domain, Petrinets for Formal Verification
- In Chapter 3, the Methodology for Scenario based formal verification of Software Architecture is presented.
- Chapter 4 presents the Smart Learning Environment Reference Architecture with Blockchain, Design of the Examination Scenario, Flow chart for the Examination scenario. It also discusses the generated Petri net for this Scenario followed by Petri net Simulation and Results, including Behavioral Properties evaluation.
- In chapter 5, we propose a Smart Learning Environment Reference Architecture with Interoperability Engine. The Workflow of the Interoperable engine functioning is presented with the generated Petri net for the above scenario, followed by analysis of simulation and Behavioral properties analysis.
- Chapter 6 presents Smart Learning Environment Reference Architecture with Personalization, workflow of the Personalization as a service, and realization of the proposed pipe-filter heterogeneous architecture with a pertinent followed by simulation and behavioral properties analysis.

1.5 List of Publications

- Published and Accepted

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. [Book chapter]

2. G R Anil, Salman Abdul Moiz, Formal Verification of a Block chain-enabled Smart Learning Environment Framework using Petri NETs"", *International Journal of Future Generation Communication and Networking* Vol. 13, No. 2, 2020 pp.838-853 [Index: ESCI]

3. G R Anil, Salman Abdul Moiz, "Personalized Dynamic Learning Plan Generator for Smart Learning Environments", *International Journal of Recent Technology and Engineering (IJRTE)* ISSN: 2277-3878, Volume-8 Issue-2, July 2019. [Index: SCOPUS]

4. G. R. Anil and S. A. Moiz, "A holistic rubric for assessment of software requirements specification," 2017 5th National Conference on E-Learning & E-Learning Technologies (ELELTECH), Hyderabad, 2017, pp. 1-6, doi:10.1109/$ELELTECH$.2017.8075001. [Index: SCOPUS]

Chapter 2

Literature Survey

2.1 Smart Learning Environments

The world is getting smarter with the Smarter in each field with emerging technologies that performing tasks efficiently and effectively. The Online learning environments are no excuse from getting smarter. The International Association of Smart Learning Environments (IASLE)[30] has defined smart learning as: "an emerging area alongside other related emerging areas such as smart technology, smart teaching, smart education, smart-e-learning, smart classrooms, smart universities, smart society. The challenging exploitation of smart environments for learning together with new technologies and approaches such as ubiquitous learning and mobile learning could be termed smart learning" [25] [27].

As the Educational technology domain has amalgamate approach from Educational theories and Technological approach, a thorough analysis of requirements has to be performed. The detailed requirements are emphasised by some Standard bodies as well as the Educational pioneers and research communities. They are presented in the following sections. The literature for this domain can be divided into two parts. Theoretical conceptualizations and the Technical specifications through designing the learning environments.

International Standards Organization(ISO)'s IT standard for Learning, Teaching and Training (ITLET), a group that works on standardization in learning environments has specific

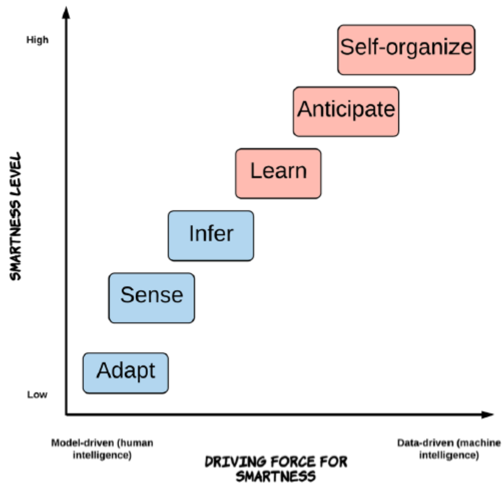


FIGURE 2.1: Smartness levels in the Smart Learning Environments

set of challenges. Tore Hoel and Mason [27] examined the outputs from the ITLET standards research community and Education Research Community to identify how evolving conceptual frameworks could overcome the limitations of old frameworks struggling to map emerging requirements with the new dynamics.

2.1.1 Smartness levels

Its a challenging task to measure the Smartness in a learning environment. Uskov et Al [62] has categorized the new age requirements from Standard bodies and Research Community into 6 levels. Each level is balanced with certain Standardization Challenges, Technical activities and classroom activities. The six levels proposed namely are Adapt, Sense, Infer, Learn, Anticipate, Self-Organize.

It can be observed that the first level is more of Human Driven(Human Intelligence) leading towards Data-driven(Machine Intelligence) at Last level(Self-Organize). This can be visualized from the diagram 2.1.

The standardization challenges in each level are tabulated in figure 2.1.

SLE levels	Smart Classroom Activities	Technologies involved	Standardization challenges
Adapt Ability to modify physical or behavioral characteristics to fit the environment or better survive in it.	• Communicate (local and remote) • Share content • View content in a preferred language • Initiate session with voice/facial/gesture commands • Ask questions • Present (local & remote) • Discuss • Annotate	• Web technologies • Session-based analytics • Personal digital devices • VR and AR systems • Presentation technologies (Smartboards, etc) • Social media • Sensors (air, temperature, number of persons, participation roles, ...)	• Setting up a SLE meeting quality criteria defined in Smart Classroom standards • Data governance • Privacy • Security • Systems interoperability
Sense Ability to identify, recognize, understand and/or become aware of phenomenon,event, object, impact, etc.	•Automatic adjustment of classroom environment lights, AC, temperature, humidity, etc. • Real-time collection of student feedback from diverse contexts • Monitoring student activity • Process real-time classroom data • Deliver custom support and scaffolding for special needs students • Support agent-based systems • Interact with smart systems • Connect multi-location students	• Triggers actions, defined in assorted models (learner, school, teacher, Smart Classroom, etc.) • Big Data • Multiple interfaces and channels keyboard,screen, voice, agent, eye movements, gestures	• Data collection and storage • Data governance • Privacy • Security
Infer Ability to make logical conclusion(s) on the basis of raw data, processed information,observations, evidence, assumptions, rules and logic reasoning	• Recognize every individual • Process real-time classroom data • Process incomplete classroom data sets • Discuss presented learning content and assignments with remote students in real-time in their preferred language	• Simple rule-based process engines • More complex inference engines • Natural language processors	• Pedagogical designs • Student learner models • Student activity data • Specifying competence

TABLE 2.1: Smartness levels in Smart Learning Environments derived from Uskov[62] Part-1

SLE levels	Smart Classroom Activities	Technologies involved	Standardization challenges
Learn Ability to acquire new or modify existing knowledge, experience, behavior to improve performance, effectiveness, skills, etc.	• Ability to suggest changes to the system • Real-time skills assessment • Real-time knowledge assessment • Accommodate and enact multiple intelligences	• Artificial Intelligence • Machine Learning • Deep Learning	• Validating competence • e-assessment • Learning Design
Anticipate Ability of thinking or reasoning to predict what is going to happen or what to do next.		•Predictive engine (predictive analytics)	
Self-organize Ability of a system to change its internal structure (components), self-regenerate and self-sustain in purposeful (non-random) manner under appropriate conditions but without an external agent/entity.		• All above, with a strong AI component	

TABLE 2.2: Smartness levels in Smart Learning Environments derived from Uskov[62] Part-2

With the introduction to Smartness and Smartness levels let us look into the existing Standards and their smartness.

2.2 Standardization and Advantages

Standardization work involves conceptual, technical, political activities that focus on achieving consensus among groups of stakeholders. An approved standard is typically a document that works as a reference point, including some technical specifications. As the requirements are evolving continuously, the standardization must focus on the concurrent requirements too. Domain experts and the Standardization community always ask for new frameworks for development.

Developing the "Specifications" is the most significant part of the Standardization process. Specifications are like requirements, which need to be presented before the standardization process begins. Unless approval from the Standards development bodies, they are considered incomplete or experimental [22]. Standards capture a strong agreement between the development body members and organizations. With such a complex process, standards evolve slowly. An approved standard is highly conclusive and considered to be complete. To be definitive and complete, they must capture wide acceptance, serve general purposes, and be prepared to manage the long-term risk([22]).

The Smart Learning requirements can be majorly considered from the ISO/IEC ITLET(IT Standardization for the Learning, Teaching and Training).

Some standards are Reference models, and some are Reference Architectures. A reference model consists of the functional division of components and the interactions between the components. A reference model portrayed into Software elements is called a Reference Architecture. In this mapping, the functionalities of the reference model are collectively represented with components, and the data flow between them is also defined. A reference model decomposes the functionality, whereas a Reference architecture maps the decomposed functionality. The relation between Reference models, Reference Architectures, and Software Architectures is mentioned in Figure 2.2.

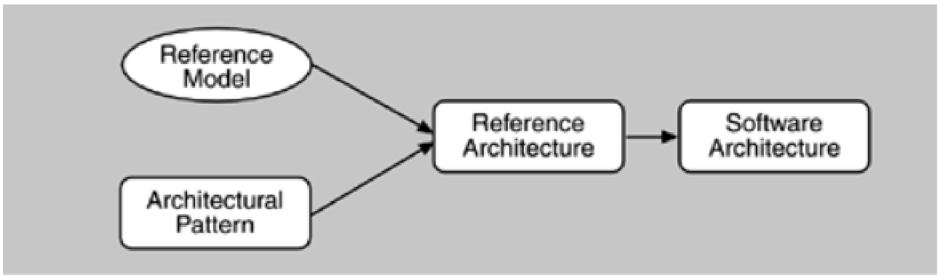


FIGURE 2.2: Relation between Reference model, Reference Architecture, Software Architectures

The Reference Architectures provide an outcome of early design decisions. Len Bass[10] stated that "A software architect must design a system that provides concurrency, portability, modifiability, usability, security, and that reflects consideration of the trade-offs among these needs." The important contribution from the Research work on Reference Architectures is that a set of multiple Software Architectures can be derived with varied specifications. Each outcome can be analyzed with early design decisions and proceed with the optimal or sufficient resultant architectures. The early analysis enhance the quality of the systems in several aspects.

2.3 Standardization works in Learning Environments

The standardization work can be divided into two sections, Theoretical conceptual models and Technological models.

2.3.1 Conceptual Standardization

The theoretical conceptualizations represents the decomposition of the functionalities, they are informative. They provide the required functionalities, but not the path for Software Architectures. They are from the Educational domain, which consider the requirements of standard bodies. The following are the significant frameworks which refer to the ISO/IEC SC36 requirements and other bodies.

Spector [60] (2014) focuses on "conceptualizing the emerging field of smart learning environments" pointing to "three foundation areas that provide meaningful and convergent input for the design, development and deployment of smart learning environments: epistemology, psychology and technology". In his work, we can note the list of requirements

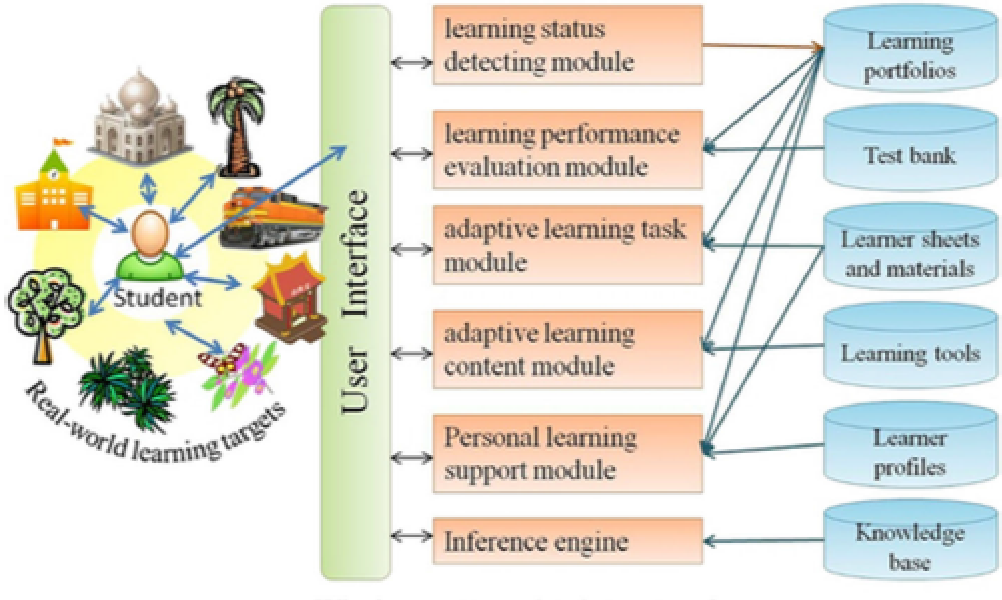


FIGURE 2.3: Hwang Framework for Smart Learning Environments

and their categorizations for the Smart Learning Environments. The personalization is placed in the 'highly desirable' category and the efficient, effectiveness are mentioned in the 'Necessary' category. The first three Smartness2.1. levels also emits the same conclusion.

Hwang's framework [29](2014) for Smart learning is little more informative. Learner Status detection module is one of the functional requirement mentioned in this level, that is reflected in any definition of the Smart learning, which we have tried to address as a scenario in the chapter 6. Hwang's framework is depicted in the figure 2.3

Koper [33] 2014 defines SLEs as "physical environments that are enriched with digital, context-aware and adaptive devices, to promote better and faster learning". Koper emphasised that the smart learning devices must be information rich and also monitor the learning process.

Zhu [68] et Al (2016) highlighted ten key features of a Smart learning Environment which includes Interoperability, Context-aware, location aware, seamless connection etc. In our work, we have addressed the interoperability challenges in the Chapter 5 and specified the location details in Chapter 4.

Liu et al. (2017)[37] proposed a Learner centric model. This model has four levels: Learning Scenarios, Learning experience, Teaching principles & learning, and support technologies. All requirements are covered at various levels in Smartness standardization challenges.

These conceptual models represent contemporary requirements theoretically. They must be addressed technically in order to implement and utilize in real time.

2.3.2 Technical Standardization

The Technical frameworks for the Smart Learning Environments are lagging behind the conceptual models. While designing the learning systems, the technical models give more inputs. The technical frameworks in the literature are pretty older and new standards need to be developed to address current requirements[27][25]. As the technologies are getting more sophisticated, there is clear need for adopting them to the learning environments. The frequent upgrades is a limitation for the technical frameworks in learning environments. Thus, ability of the learning systems for a easier modifiability and adaptability need to be enhanced. This can be achieved by reducing the dependability among the components of the learning system. by performing modifications in one component, must show least impact on other components in the learning system.

The technical frameworks are proposed by various standard bodies throughout the world.

- IMS abstract framework :

  IMS Global Learning Consortium proposed IMS abstract framework in 2003[16]. In their inception, IMS described their project as "defining the internet architecture for learning"[27][59].

  The IMS abstract framework is a tool that describes the context within which IMS develops its learning systems' technical specifications. The IMS Abstract Framework is so named because

  1. In this framework, the broad set of e-Learning services are abstractly presented.
  2. Supporting various distributed electronic learning systems is focussed in this framework

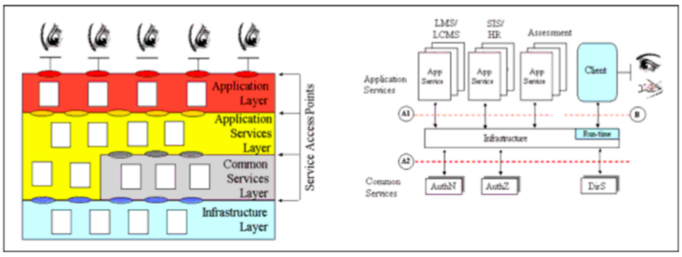


FIGURE 2.4: IEEE standard for Learning Technology Systems Architecture.

3. This framework covers various e-Learning architectures that are constructed on a collection of defined services.

This IMS framework can work as a starting point for designing various e-learning systems. Any learning system can use a portion or entire parts of this framework's specification for instant use. The process of a particular learning system implementation through a defined architecture or reference model is called "Profiling of the framework" by IMS GLC.

There are four layers in this IMS framework namely, Application layer; an Application services layer; a Common services layer; and an Infrastructure layer. The layered structure of the IMS framework is shown in the figure 2.4.

The IEEE LTSC has many input channels; IMS is one among them. For the upcoming standards, IMS may or may not be considered by this standard committee. But, IEEE LTSA has adopted the IMS's metadata and approved it to be a standard.

Since this framework is too abstract, it would be difficult for the developers to understand the system. Thus proper decomposition of the system into meaningful components might fail. This is the major limitation of any abstract framework. [27]

- OKI In 2002, the Massachusetts Institute of Technology (MIT) had taken the initiative for Open Courseware to the world. This OKI project has initiated defining ITLET architectures with a service-oriented approach. Also, it used Open Service Interface Definition(OSID) as programmatic interface specifications for describing services. These interface definitions support interoperability across platforms developed with different technologies. Given the subsequent revolution in cloud services that rendered many enterprise architectures redundant, OKI made inception for this

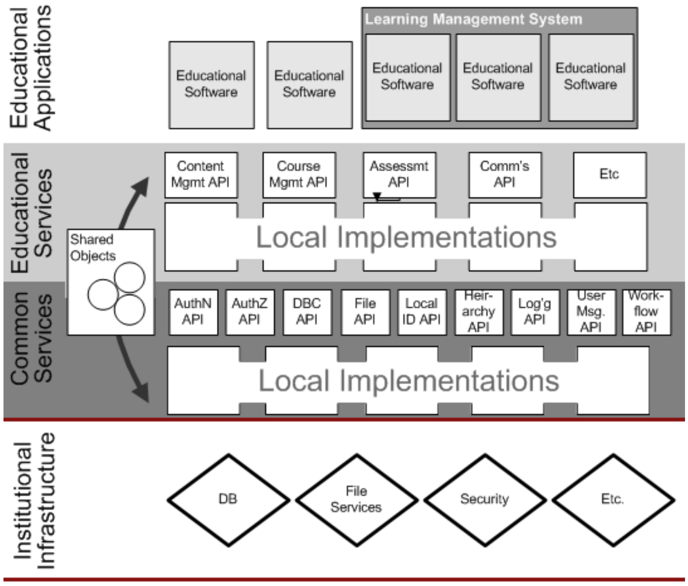


FIGURE 2.5: O.K.I Architecture

revolution. However, MIT has discontinued this project and did not maintain its archive on the OKI website. It has conveyed a strong message that innovation in digital infrastructure is subjected to disappear in a shorter period. But, the OKI initiative has reached beyond learning in this domain. It explicitly attempted to achieve knowledge along with learning. At that time, there was a rich emergent discourse that articulated the notion of shared services between knowledge-based systems and learning [42] .

The OKI architecture is shown in the figure 2.5

- JISC The UK's Joint Information Systems Council (JISC)[19] is an ICT support agency for a service-oriented view of e-Learning in Universities. There are three categories of services in this standard; they are Learning domain services, Common services, and Simple User agents. JISC developed a framework to "enhance learning by creating an open programming environment that supports sharing and pedagogical experimentation" (JISC, n.d.). JISC has become the precursor to the multinational collaboration on the e-Learning framework for Education and Research. It was funded by government agencies from the United Kingdom, Australia, Netherlands, and New Zealand. The ISO/IEC/TS 20013:2015 standard considered JISC as a useful reference framework of e-Portfolio information in 2015.

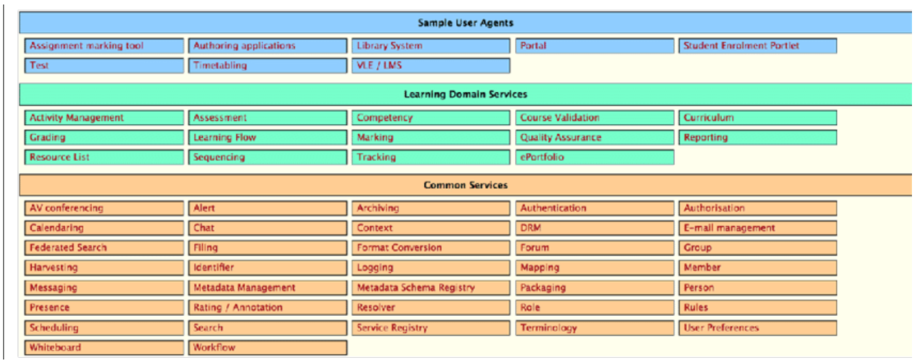


FIGURE 2.6: JISC E-Learning Framework

- ADL – The Total learning architecture

  The Advanced Distributed Learning (ADL)[36] had developed the Sharable Content Object Reference Model (SCORM) [5]. The ADL is a United States Department of Defence (DOD) program for data interchange. Later, ADL has developed the "Total Learning Architecture." (TLA) [36]. It considered the advanced cloud services and data-rich contemporary environment. SCROM was designed to mention the real-time requirements of a self-paced learner's individual sessions maitanance in a learning environment. ADL has developed xAPI (the eXperience API) to store the learners' individual experiences, learning outputs, etc. The xAPI is more sophisticated than SCORM in terms of features. The TLA(Total Learning Architecture) is very near to the Smart Learning Environment conceptually. However, it cannot be a Reference Architecture as it is conceptual.

2.3.3 IEEE Learning Technology Standards Committee

Apart from the above conceptual frameworks, the widely accepted standard for developing an e-Learning system is "IEEE Standard for Learning Technology Systems Architecture."

IEEE Computer Society Standards Activity Board charters IEEE Learning Technology Standards Committee (LTSC) to develop globally accredited technical standards, suggested practices, and guidelines for learning technology[39].

The IEEE 1481.1-2003 standard of Learning Technology Systems architecture is a high-level architecture for information technology-assisted learning systems. This standard describes its high-level system design with specified components. It is the purpose build

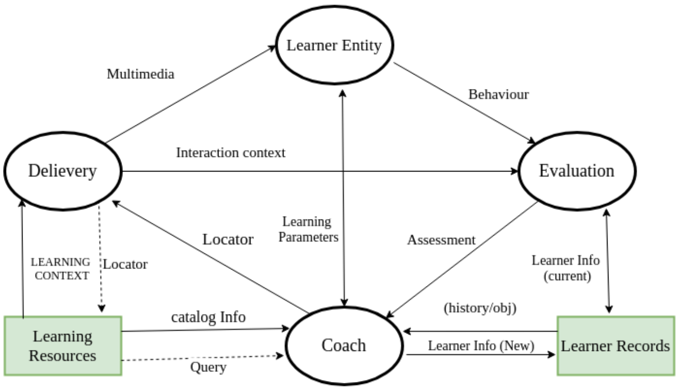


FIGURE 2.7: IEEE standard for Learning Technology Systems Architecture.

learning technology standard that served as a stable reference architecture. It provides a framework for understanding the existing and future systems.

The Learning Systems Technology Architecture (LTSA) is a widely known architecture with abstract components. The higher-level abstractions can be implemented in lower levels, either as actual implementations or lover-level abstractions. The validity of this standard is mentioned as, "This standard has a technical horizon or applicability for 5 to 10 years from release while adaptable to new requirements, technologies, and learning management systems. "[39]

An architecture standard does not represent a single system. It is a generic architecture with which a range of systems can be developed. With the right amount of generality, the components and sub-components of the systems can be developed separately. Such developed components are reusable, cost-effective, and adaptable(i.e., identifying abstract, high-level interfaces and services. Based on this IEEE LTSA standard, any architectural framework derived does not regard the particular implementation details necessary to create learning technology system components.

Th high level architecture of the IEEE LTSA is depicted at Figure 2.7

The components of the LTSA are presented in the table 2.3

Learner Entity

- The Learner Entity process is the reflection of a human learner. It can just be a learner, a set of learners learning collectively or individually.

Table 2.3: IEEE LTSA Components

Component Type	**Component Name**
Processes	learner entity, evaluation, coach, delivery.
Stores	learner records, learning resources.
Flows	learning parameters, behavior, assessment information, learner information (three times),query, catalog info, locator (twice), learning content, multimedia, interaction context.

- Input to the Learner Entity is "Multimedia," which represents the learning content. The Behaviour of the learner is observed and sent to the Evaluation process. Multimedia and behavior are abstractly described terms. There can be an implementation of multiple human interface modules like Windowing systems, Web presentation modules, discovery laboratories, etc.

- The Coach process negotiates with Learner Entity with the Learning Preferences.

Coach :

- The Coach process interacts with the Learner Entity through the Learning Preferences connector. The Learning preferences can include learning strategy, learning style, etc.

- The Evaluation process submits the Assessment information to the Coach process. This information helps the Coach process to support the decision-making process.

- The Coach selects the appropriate learning content from Learning Resources through queries based on the latest assessment and previous performance. The Learning Resources responds to the queries and returns Catalog Info. The Catalog info consists of the Learning Content index from Learning Resources.

- The suitable content Locators like lesson plans are sent to the Delivery process.

Delivery:

- The Delivery process derives the Learning Content index(Locator) from the Learning Resources. The selected Learning content is converted into a Multimedia presentation.

- The Coach process shart the Locators share Locator with the Delivery process. The Deliver process access LEarning content from Learning Resources. There can be any type of delivery mechanism like Teacher defined, or Student opted, etc.
- To maintain tight coupling, The Delivery process can be implemented with the Evaluation process. This tight coupling can help for an enhanced learning experience.

Evaluation:

- The Evaluation process retrieves Behavior from LEarning Entity and produces Assessment information and Performance.
- This Evaluation process transfer the "Assessment Information" to the Coach. The Performance is interchanged and stored at Learner Records.
- The Evaluation process uses the "Interaction Context" and Learner Entity's "Behavior" to determine the appropriate Evaluation. For example, a Learner Entity is expected to select from a multiple choice question and the correct answer is "# 2", but the Evaluation process has no context that the keystrokes "2" (or "# 2" or "two") are the correct answer the Learner Content provides the context ("the correct answer is # 2") to correlate the correct Behavior (e.g., the keystroke "2").

Learner Records:

- Learner Records is a data store. It stores Performance details of current and past and also future objectives.
- Learner Records collects Learner information from Evaluation and gets the Performance from the Coach.
- The performance information is derived from the Evaluation and Coach process. The Coach process returns the performance information like certifications, and Evaluation returns data like grades.

Learning Resources

- Learning Resources is a database that stores the learning content that can include presentations, lecture videos, notes, experiments, etc.

- The Learning Resources data store represents the "knowledge" information along with Learning content. The Learning content is accessed through queries. The matching information returns Catalog information. The catalog information is the learning content index that stores the list of entire learning content. This list or Catalog is a set of learning content tags that are also known as learning object metadata. The Locators facilitate the Delivery process to access the appropriate content from the Learning Resources. For example, a query on a topic in chemistry (specified as a Query) might return a set of Catalog Info that include a laboratory experiment simulating the behavior of solids, liquids, and gases, a presentation on Boyle's Law, a bibliography, or related materials, a tutorial, a chemistry tutor (the tutor may be human or surrogate; geographical nearness is irrelevant), and an ontology (a conceptual model of the subject represented as generic Learning Content) for temperature.

2.3.3.1 Conformance to IEEE standard for LTSA

A Learning system is conformance to the IEEE LTSA in three ways of adopting the components [39].

- Conformance with all components in the standard
- Conformance with some components but not all from the Standard
- Conformance with components except few components from the standard

Any proposed architecture is conformance to IEEE standard for IEEE LTSA with Re-use of all components or at least some components.

- Limitations of LTSA

  The IEEE standards committee has withdrawn LTSA because of certain limitations. This standard proclaimed itself to have validity for 5-10 years unless affirmative to the latest technologies and fulfilling the latest requirements. If a standard is not affirmative, it does not represents state of art. No new standard on Learning Technology Systems Architecture is available after the withdrawal. There are invitations to the volunteers who can join the Standards development committees to understand

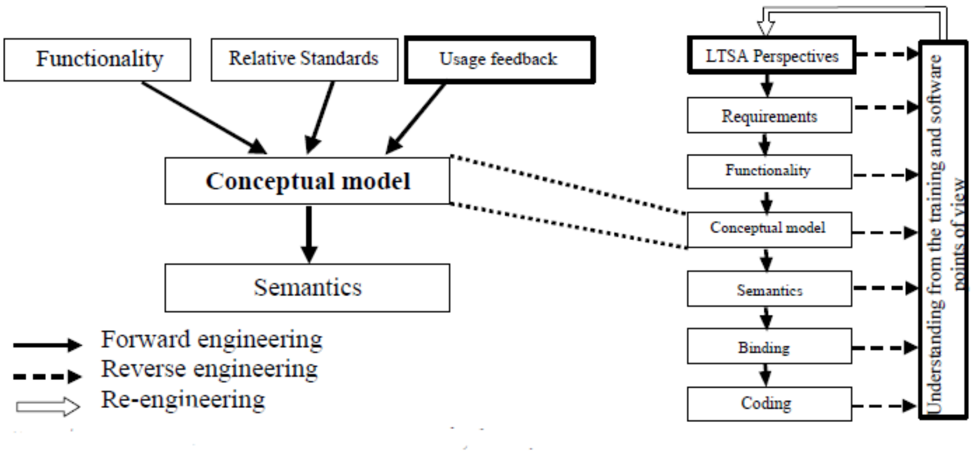


FIGURE 2.8: Workflow model from IEEE Standard to a Software Architecture

and develop new standards. There exist limitations, such as some functional areas are missing and could not achieve the current complex requirements[34]. Thereby making LTSA not suitable for Smart Learning environments. IEEE LTSA is at the Pre-Smart level of Smartness in a learning environment[27].

2.3.4 Extension Research work on IEEE LTSA

The Software Engineering process initiating from the IEEE LTSA towards developing a Learning System is explained by Alain et Al[17] through design integration interaction cycles. The development of a learning system from Reference models while addressing the requirements with Reverse Engineering, Re-Engineering, forward engineering are presented in Figure 2.8. This work highlights the Reverse Engineering till the set of requirements fulfilled. Though the smartness levels were not defined by the time of [17] proposal, it gives a holistic picture of the IEEE LTSA adoption process for learning systems software life cycle. In the 2.8, the workflow model for development of a learning system begins with LTSA perspective and it is addressed with the Requirements, Relative standards and Usage feedback. Based on requirements fulfillment, this extension procedure proceeds ahead. After each requirement addressing, a conceptual model gets generated. The conceptual model mentioned in this refer to the technical model i.e. a Reference Architecture.The generated conceptual model serves as a reference point for addressing that particular requirement.

Though IEEE LTSA standard is withdrawn, significant work is done using this standard. It have given directions and strong inputs for the learning systems designers. Researchers and educationalists from Standards Research Community has published their work. Some of their extension work is presented below, the holistic literature work is presented in the table 2.4.

Vos Kamp [63]'s "Extending IEEE LTSA" is one of the first works on the IEEE LTSA. There is a modification performed on the architecture. An additional component called "Expert" is added parallel to the "Coach" process. The Learners' performance is transmitted from Learner Records to Learning Resources while giving didactic information to the 'Coach' process. The extended architecture is not validated, and no Smartness challenge is addressed.

Mairtin S. O'Droma et Al [46] used the IEEE LTSA as a reference model for the e-learning Virtual University Information system. Some functionalities are compared with 10 Learning Management Systems available in the market. This comparison highlighted the inadequacies of the existing products. The proposed architecture or IEEE LTSA are not formally represented and its Normative in nature. The Smart standardization challenges are not being discussed.

Claus Pahl et Al [14] mentioned two enhancements to LTSA such as metadata annotation and learning object assembly to be a better Web Services Architecture. But there is no Architecture mentioned in this work.

George Yee [67] presented the Context-aware Privacy and Security agents in the IEEE LTSA. The necessity of Privacy and Security in the learning domain is highlighted. It is one of the significant Standardization Challenge in the First and Second levels of Smartness too[62]. In George Yee et al.'s work [67], the IEEE LTSA reference architecture is prescribed with the privacy and security agents at each component.

Three Scenarios from previous literature that are " Coach to Evaluation process interaction," Coach to Learning Entity interaction, Learning Entity to Coach interactions, have been carried out with the help of the pre-defined PS agents(Privacy and Security agents).

But neither the architecture nor Scenarios are formally described and validated. The considered scenarios are based on interaction content, and they did not specifically mention a function of the learning system.

Sengupta [56] attempted to propose a hybrid model with the IEEE LTSA. This model is based on IEEE LTSA, and did no modifications were done to the architecture. The learning is quantified with Learning quanta specifications with few specific features. This contact learning(part of the hybrid model) is presented with the UML notations such as use-case, Sequence, and Activity diagrams. With the generated UML diagrams, the implementation of a Learning Management System can proceed. With the same IEEE standard for LTSA, the system can achieve the quantification of learning. A much more early analysis before implementations would have been easier if expressed in Architecture Description Languages.

Francisco [23] proposed a Service-based framework for the Personal learning environments in Learning management systems. There is no architectural model presented in this work. Some personal learning environment features are introduced. These personal environment characteristics are part of Smartness Standardization challenges too.

P. Kumar [34] et al. presented a detailed extension for IEEE LTSA into a secure Client: Server model and towards a Distributed model. One of the significant limitations of the IEEE LTSA is being a centralized database system. This limitation is addressed in this work. Some inputs have been taken from this work while designing distributed blockchain architecture with IEEE LTSA. Their proposal is analyzed with a case study, but architecture has not been validated.

Yingchun Yian [61] has proposed a learning system for Music by adopting IEEE LTSA driven with an ontology based approach model. Its a kind of application developed referring to a standard.

Faisal et al. [26] has proposed Architectures for learning based on IEEE LTSA. They have extended IEEE LTSA with 4+1 View model (one of the architectural patterns). As discussed at Relation between Reference model, Architectural pattern and Reference Architecture 2.2, this work extended to get a specific architecture by mapping the levels in IEEE LTSA with levels in 4+1 view pattern. This system is represented with a Component diagram, Process diagram, Package diagram, and Deployment diagram. It can be further extended formally as described in Chapter 3 and perform early analysis.

Andharini et al. [7] has designed an adaptive E-Learning application by referring to IEEE LTSA. The functionalities for adaptive learning are mentioned, and they are presented through a Use Case diagram and Class diagram. There is no architectural modification done to the IEEE LTSA. But, this work set an example for designing an adaptive learning environment. It is one of the advantages of a Reference model that can extend according to the specific requirements.

Isiaka et al. [32] proposed the Web based Instructional Design by extending IEEE LTSA. Here, the Coach process is extended to accommodate pedagogical content packaging. In this design, three actors are defined: Lesson designer, Content author, and lesson tutor. This architecture is not expressed with architectural description language and is not validated.

The extension works on IEEE standard for LTSA are presented against the parameters like Compliance with IEEE LTSA, Architectural modifications, Formal validations, Smartness challenges addressing in Tables 2.4, 2.5.

The works from Smart Learning Environments are presented in Table 2.6. These are the informative frameworks that present the Smart learning environments conceptually(i.e., informative). These are some of the latest theoretical works from the International Association for Smart Learning Environments(IASLE). These are not Software Architectures with which one can proceed implementation; rather, we can consider for designing a Smart Learning system.

Spector [60] has proposed a Preliminary framework for a Smart learning environment with categorized features. The features(requirements) are from multiple domains like Psychological, Physiological and Technological. There are three categories for features: Necessary, Highly Desirable, and Likely. The Necessary category features are mandatory, Highly desirable are required, and Likely are optional but advanced.

Name	Year	Workdone	Compliance with IEEE LTSA	Architecture Modification	Description with ADL	Formal Validation	Built of any Scenario	Smartness Challenges
M S O'Drama	2003	"Virtual University Information System" with IEEE LTSA	YES	NO	NO	NO	NO	NO
Claus Pahl	2004	Web Service Architecture with metadata and learning object assembly	YES	NO	NO	NO	NO	NO
George Yee	2005	Context aware Privacy and Security agents in IEEE LTSA	YES	YES	NO	NO	NO	YES
Senguptha	2009	Hybrid model with Learning Quanta with IEEE LTSA	YES	NO	NO	YES	NO	NO
Fransicso	2011	Service-based framework for Personal Learning Environments	NO	NO	NO	NO	NO	NO
P Kumar	2010	Extended LTSA in SOA environment	YES	YES	NO	NO	NO	NO

TABLE 2.4: Extension works on IEEE standard for LTSA Part-1

Name	Year	Workdone	Compliance with IEEE LTSA	Architecture Modification	Description with ADL	Formal Validation	Built of any Scenario	Smartness Challenges
Yingchun Yian	2010	Learning system for a Music College adopting IEEE LTSA	YES	NO	NO	NO	NO	NO
Faisal	2013	Mapping of Software Engineering models with IEEE LTSA	YES	NO	NO	NO	NO	NO
Andharini	2014	Designed an adaptive E-Learning application	YES	NO	NO	YES	NO	NO
Isiaka	2016	Web based Design to accommodate the pedagogical content packaging	YES	YES	NO	NO	NO	NO

TABLE 2.5: Extension works on IEEE standard for LTSA Part-2

Hwang [29] proposed a conceptual framework that considers the key criteria for Smart Learning Environment: context-awareness, adaptive, and ability to adapt user interface, subject content, and reporting learning status.

Koper (2014)[33] defines SLEs as "physical environments that are enriched with digital, context-aware and adaptive devices, to promote better and faster learning". Koper considers the digital devices at the learner location have impact on learning. This framework highlights the location aware, context aware, Culture aware system. Koper has identified 5 levels of Human Learning Interfaces with cognitive science perspective.

Zhu et al [68] proposed a Smart Learning Framework, along with defining a ten significant features. The features are location-awareness, context-awareness, interoperability,adaptability, seamless connection, social awareness, ubiqui- tousness, Complete learning record including learning paths, natural (multimodal) interaction, and high engagement.

Name	Year	Compliance to Smart Learning Environments	Architecture proposed	Formal Validation	Normative or Informative
Spector	2014	YES	NO	NO	Informative
Hwang	2014	YES	NO	NO	Informative
Koper	2014	YES	NO	NO	Informative
Zhu	2016	YES	NO	NO	Informative
Liu	2017	YES	NO	NO	Informative

TABLE 2.6: Smart Learning Environment Frameworks from [60][29][33][68][37]

Liu [37] proposed a conceptual framework with four levels. They are Learning experience, Learning scenarios, basic principles of teaching and learning), and Support technologies. It is a learner-centric model.

2.3.5 Summary

In this chapter a comprehensive review of the existing work in the Learning Technology systems architectures and Smart Learning Environments is studied and reported. The architectures extended on IEEE LTSA in different perspectives are reported. Based on the literature review, the following research gaps reported.

- The extension of a standard towards addressing ITLET requirements, smart Learning Environments standardization challenges is not carried out, which motivates to proceed with it.
- As presented in the 2.6, the extension works of IEEE LTSA are not attempted formally with representations like Architecture Description Languages. To the best of our knowledge, this is the first of its kind.
- The IEEE LTSA is not extended in order to address the Smartness challenges.
- As presented in the 2.6, the latest research work in SLE is dealing with the Conceptual models(informative), there is a research gap to proceed with the Architectural design of these conceptual models(normative).
- As per the recent survey paper [27] from the journal "Smart Learning Environments," and the Standardization challenges for the Smart Learning Environments need to be

addressed with new frameworks. This conclusion gives the motivation to proceed with the Reference Architectures as an initial contribution for new standards

- Extending a standard to address Smartness challenges can make the standard reach higher level smartness. The architectural validation of system designs makes a significant contribution to system development. Formal verification of the architectures is not addressed in this area, which motivates to experiment in this approach.

Chapter 3

Methodology for Reference Architectures Validation and Scenario based Verification

This section describes the methodology to express the proposed architectures formally followed by Architectural Validation and Scenario based Verification. The overall procedure for Architecture verification is described in the Section 3.1.

In order to achieve the higher Smartness levels, we proceed with addressing the Smartness standardization challenges. The extension of the IEEE LTSA architectures requires architectural modifications which results in new architectures. The validity of the proposed architectures are performed by describing the new architectures with Architecture Description Languages and their supporting tools. Scenarios that utilize the newly added/-modified components of the IEEE LTSA are designed to asses the workflow. A Procedure for Scenario based Formal Verification of Software Architectures are mentioned in Section 3.2.

The above procedure is evaluated in two phases.

- Architectural Validation
- Scenario based Verification

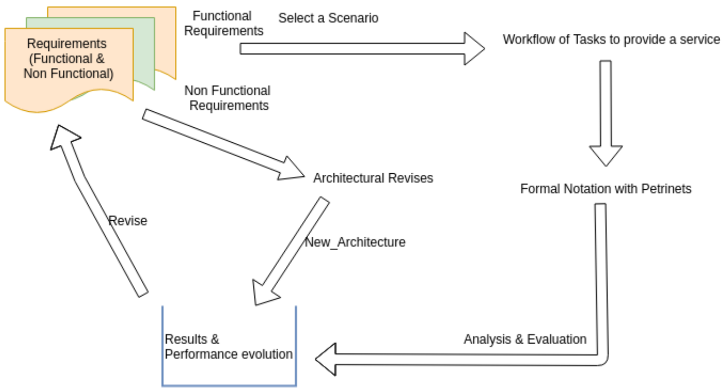


Figure 3.1: Procedure of Verification of a Software Architecture[64]

3.1 Architectural Validation and Experimental Setup

Th Architectural Description Languages(ADL) are formal languages that represents the Software Architectures. ADLs are the modelling languages that constitute set of rules and common practices. Every ADL has a precise syntax and semantics that can be expressed with the associated tools. These tools aid the creation, use and analysis of its models and visualizations. Some of the Architecture Description Languages [43] in practice are xADL, ACME, AADL, Koala, AUTOSAR etc.ADL tools can perform Automated dedicated analysis followed by Validation of the architecture against syntax, semantics and simulating systems behavior.

For this research work "Architecture Analysis and Description Language" is chosen for Architecture Description Language. It is a standard published by the Society of Automotive Engineers as reference AS5506B. This language is based on Component and Connector paradigm. Component descriptions are used for functional description of components such as data, event inputs, outputs etc. The Connectors has precise operational semantic characteristics that are enforced by AADL modelling environment. AADL is a large modelling framework. OSATE is the tool that supports AADL. OSATE abbreviated as Open Source AADL Tool Environment. OSATE has several modelling capabilities. OSATE can model the syntax aware modelling and associated with Graphical editor as well. This tool can generate the Architectural diagrams for the code written in AADL. The validation of the generated diagram ensure the semantic checks and usage of appropriate modelling properties. This tool display errors and warnings on validation. In this research work, we have expressed the architectures in OSATE, generated the architectural diagrams from the tool and performed validation on each architecture. The validation results without error and warning can be adopted by the learning system designers and proceed with the early analysis by incorporating specific inputs based on their business needs. The architectural code that is generated works as a reference architecture for them. After such validation, they can proceed with their optimal specific inputs. We have first validated the IEEE LTSA in this chapter, later in each chapter the proposed architectures are validated.

3.2 A Procedure for Scenario based Formal Verification of Software Architecture

For each standardization challenge an architectural modification is performed. Now, we need to verify the newly added component and its interactions with the other components. A Workflow for a particular task that engage the newly added components need to be designed to verify the architecture. The Scenario based formal verification of a framework can be performed with the following procedure. The workflow of each scenario need 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. In this work, we verify with a scenario associated with each standardization challenge. Each scenario is expressed with an activity diagram(Semi formal representation) followed by Petrinets(Formal representation) and analysed.

Generally a workflow for a service is demonstrated with the help of UML diagrams especially activity diagram. They are helpful in modeling use cases for representing business workflows. In the activity diagram, every event is taken as an activity.

To build an activity diagram for a workflow of a scenario,

- 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[53], systems are also modelled with colored petrinets[40]. A brief introduction about Petrinets is given in the section 3.2.1. 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 as a transition and Every decision as a place. From this procedure, a Petri net can be generated. A Petrinet analysis can conclude the correctness through the Safeness, Liveness(Deadlock free) and Boundedness[38].

3.2.1 Petri nets

A Petri net [2][51][58]is also known as a workflow net or place-transition net. It is a formal approach to measure the performance of the system as a bipartite graph having two categories of nodes called places and transitions[2]. There would be directed arcs that connect the places and transitions. A Petri net can be used for an effective illustration of distinctive features of a system like concurrency, decision making, etc. The dynamic performance of a Petri net designed systems is to deliberate in terms of its states and its changes. Every place may have either zero or a positive number of tokens. The basic elements of Petri nets are Places, Transitions and the tokens associated with the Places. The presence or absence of a token in a place indicates whether a mentioned condition related to this place is true or false.

A Petri net is explicitly stated as a 5-tuple $N = (\ P,\ T,\ I,\ O,\ M_1)$, where

- $P = \{p_0,\ p_1,\ p_2\ ,\ .\ .\ .\ ,p_{n-1}\}$, a collection of places in the Petri net. Generally, they are represented in the shape of a circle.
- $T = \{t_0,\ t_1\ ,\ t_2\ ,\ .\ .\ .\ ,\ t_{n-1}\}$ is a collection of Transitions of a Petri net, which generally represented with a shape of a bar or a line
- I is the set of arcs that join places to transitions. It stores the input relation for the transitions
- O is the set of arcs that joins transitions to Places. It stores the transition output relation.

- $M = \{ M_1 , M_2 , M_3 , ... , M_{np} \}$ is a set of marking. A generic marking M_i consists of the sum of tokens in each place p_i in that marking. Generally, they are drawn as black dots. M_1 is the initial marking.

A transition is entitled to mark all of its input places that hold at least hold one token. A transition gets enabled or fired by moving tokens from input places to output places according to arc weights and number of tokens in a place. A Reachability set R(M_1) is a collection of markings that are fired from an initial marking M_1 which is obtained by applying the above-mentioned rules repeatedly. A transition t_k is said to be enabled in a marking M_i if for any

$$p_i \epsilon I(t_i), M_i \geq 1 \tag{3.1}$$

A Marking M' is reachable from its previous marking M, that is obtained by firing a transition t_k is denoted as

$$(M - t_k \rightarrow M') \tag{3.2}$$

The token count needs to be presumed before and after firing a transition. It can be assured with the following linear equations.

$$M'(p_i) = \begin{cases} \text{M}(p_i) \text{ +1} & \text{if} \quad p_i \epsilon O(t_k), O(t_k) \\ \text{M}(p_i) \text{ - 1} & \text{if} \quad p_i \epsilon O(t_k), O(t_k) \\ \text{M}(p_i) & otherwise \end{cases} \tag{3.3}$$

Stochastic Petri nets (SPN) are another category of Petri nets which include timed transitions. They have an enormous ability to evaluate the performance with the delayed transition firings [2]. Generalized Stochastic Petri Nets (GSPN) is an advancement of SPN. GSPN was assisted by two categories of transitions that are

- Immediate transitions,
- Timed transitions.

In the immediate transition, the firing time of the transition is zero, whereas in the timed the firing rate of the transition is exponentially distributed. If both transitions are activated simultaneously the immediate transition gets fired first. If a marking is enabled in any transition, it is called "vanishing marking and in case of timed transition, it is called "tangible marking". The Petri net assists at design time analysis with the Validation, Verification, and Performance Analysis. The Validation checks whether the Model developed is reflecting what it intends to represent. The validation is carried out by analysing the behaviour properties of the Petri net that ensure the stability. The Verification techniques investigate the relation between specification and model, performs Reachability analysis and Structural analysis(e.g. invariant). Performance analysis consists of the key performance indicators eg. flow time and Simulation of processes eg. Markov analysis. The real-time assessment can be evaluated by verifying specific predicates on siphons, which is a structural object in the Petri net. The most significant behavioral properties are

- Liveness : A Transition is live if it is potentially firable in any marking of R(M_1). A transition is dead in M if it is not potentially firable.
- Safeness : If the place is not safe, it means that once initiated operation has not finished before the new initiation.
- Boundedness : A Place is bounded with bound k, if the token count does not exceed k in any marking
- Deadlock-free : A petrinet is deadlock free if there are no traps or infinite loops

There are several tools that perform the analysis automatically. Widely used tools for this Petri net generation and analysis are PIPE[52], CPN tools, HPsim, etc.

3.2.2 PIPE:Platform Independent Petri Net Editor

It's a tool for generating various kinds of Petri nets[52][55]. There are several analysis packages available in the PIPE tool. It has user friendly interface. It is capable of modelling Place-Transition nets as well as advanced Generalized Stochastic Petri nets. It can also generate Reachability graphs, generate Incidence matrices and markings. The PIPE tool can also analyse the performance through simulation[55].

The designed scenarios in each contribution are verified against the State space analysis i.e. behavioral properties mentioned above with the following experimentation setup.

3.2.3 Experimental Setup

PIPE is platform independent. However, the Experimentation set up details are as follows

Tool	PIPE Version 4.3.0
Operating System	Ubuntu 16.04LTS (64-bit)
RAM	8 GB
Processor	Intel Core i5-7500

The above methodolody and experientation set up is followed through out the research that is presented in the next three chapters.

3.3 Summary

In this Chapter, the Architecture Validation with Architecture Description Launguages and Scenario based verification are presented. A brief introduction to Petrinets along with its properties description is given. The explained procedure is applied for each Reference Architecture in the following chapters.

Chapter 4

Blockchain Enabled Smart Learning Reference Architecture

A Smart Learning Environment(SLE) is at smartness level-1(Adapt)2.1, if it can address the standardization challenges Data Governance, Security, Privacy, Systems interoperability, and meeting quality criteria of smart classroom standards. SLE can be at smartness level-2(Sense) by addressing Data collection and storage, Data governance, Privacy, and Security. The "Security" challenge is present in the first two levels. This chapter is to address some of the "Security" issues in the learning environments through Blockchain. This chapter is organized as follows: The Context and Motivation for this work is mentioned in the first section. The second section presents the proposed architecture, followed by Validation: The third section describes the Examination Scenario followed by Verification. The final section presents the conclusion of this chapter.

4.1 Context and Motivation

In order to address various threats[3][4] in learning environments, sophisticated data storing techniques with promising performance are needed. One of the robust data storage techniques is blockchain [13]. Blockchain technology is termed the 'Fifth Evolution' of computing. A blockchain is regarded as a novel approach to distributed databases. It is a distributed database of open records which documents all transactions or advanced events that have been executed prior to that point in time when it is shared among participating members.

Every transaction in the public record is confirmed with agreement by a larger number of members within that framework. A blockchain is a data structure that makes it possible to create a digital ledger of data and share it among a network of independent parties. Blockchains are secure by design and include sophisticated distributed computing systems with high Byzantine Fault Tolerance [13]. A fundamental property of blockchain technology, which distinguishes it from traditional database technology, is Integrity and Transparency and Immutability, auditability, fault tolerance, and trust-free operation [41]. In the case of the blockchain, modifying or deleting data is almost impossible. In this chapter, a blockchain-enabled Reference Architecture is proposed to realize the LTSA for Smart Learning Environments.

4.2 Blockchain enabled Smart Learning Environment Reference Architecture

The salient features of blockchain are integrated to basic IEEE LTSA framework. The proposed framework is depicted in Figure. 4.1

The proposed architecture 4.1 is "conformance 2.3.3.1 to the IEEE standard for LTSA", as all the processes are retained from the standard. Additional components to accommodate blockchain mechanism for educational transaction activity storage are added along with the standards' processes. The proposed reference architecture components include Processes (Coach, Delivery, Learner Entity, Evaluation, Network node) and Data Stores (Learning Resources, Learner records, Blockchain network). The connectors include Learning content, Learner info, history, assessment query, catalog information, and blocks. They specify data couples between the various LTSA components. The Network node is a physical device that is added to the architecture. This physical device is essential to validate the block that is being generated at the Learner end.

The User interaction is expressed in figure 4.2. Blockchain API helps in creation of blocks.

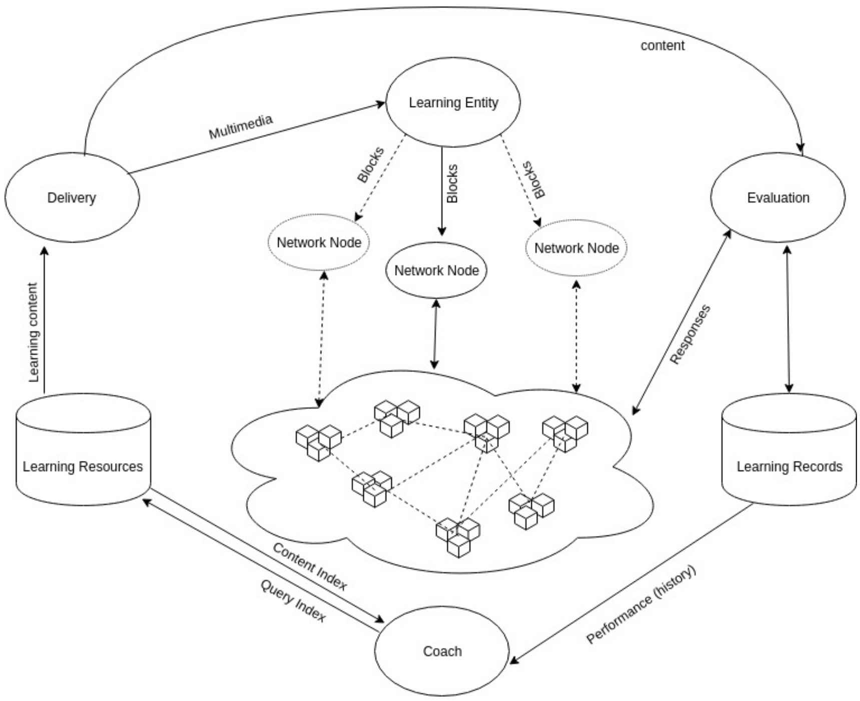


FIGURE 4.1: Blockchain enabled Smart learning Environment Framework.

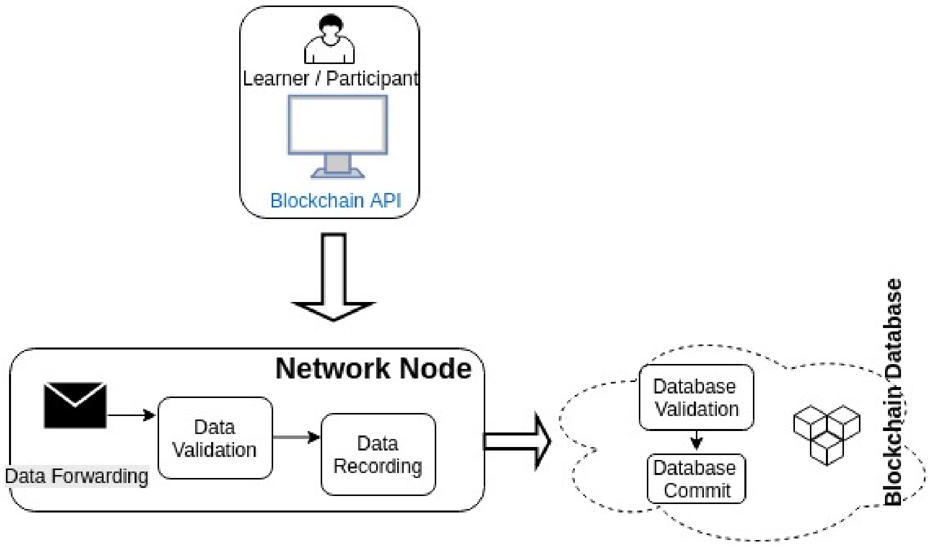


FIGURE 4.2: User interaction mechanism with Network node and Database .

The Network node performs the task of Data forwarding, Data Validation and Data recording. Once the block is validated, it will be stored in the Blockchain database(Its a distributed database). We have adopted this mechanism from Muzammal et al.' s [45] work of integrating Blockchain into a learning management system. It is a first such approach in this domain.

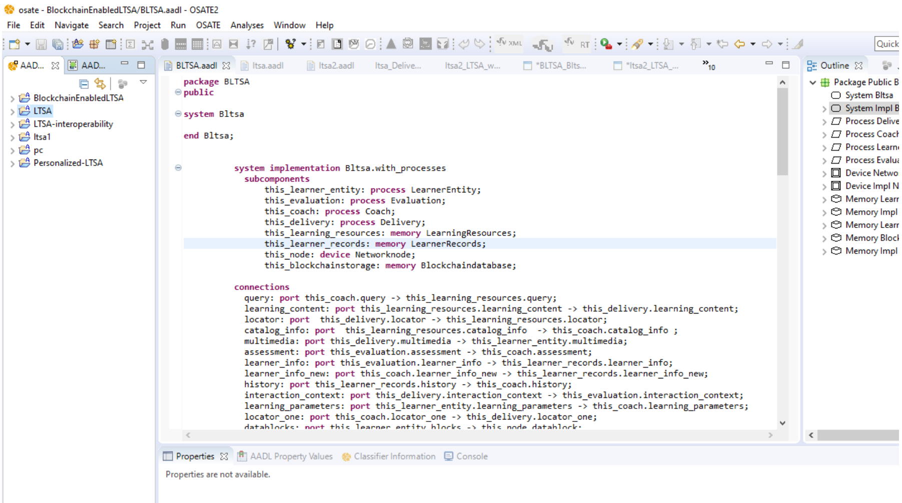


FIGURE 4.3: Blockchain enabled LTSA written in Architectural Description Language

4.2.1 Proposed Reference Architecture Validation

The proposed Reference Architecture is described with AADL architecture description language on OSATE tool. The fragment of the code is presented in Figure 4.3. Describing an architecture in OSATE is carried in terms of a project. The processes and connections need to be defined. In the proposed Reference Architecture, we have defined the processes mentioned in Figure 4.1. The network node is added as a device-type component. Except for two connectors remaining connectors of the IEEE LTSA are remained the same. The Evaluation process interacts with the Blockchain database with the connector "Responses."

.

4.2.2 Generation of Architecture diagram from ADL

The OSATE tool can generate the architecture diagram from the code for architecture description languages. The generated diagram is presented at 4.4

It can be observed that all the processes, memory, interactions are integrated properly.

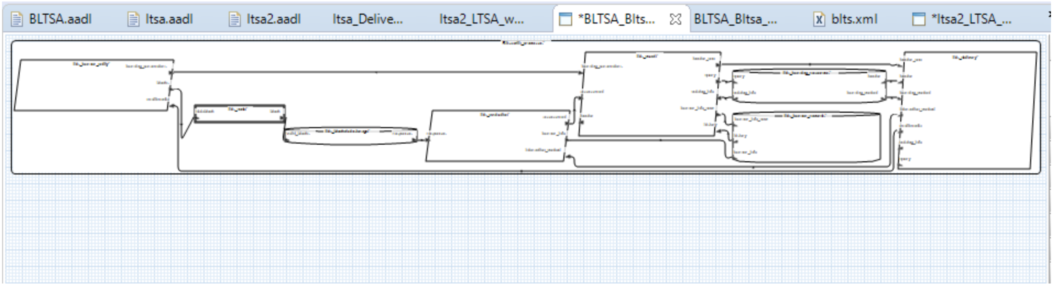


FIGURE 4.4: Diagram of Blockchain Enabled Reference Architecture

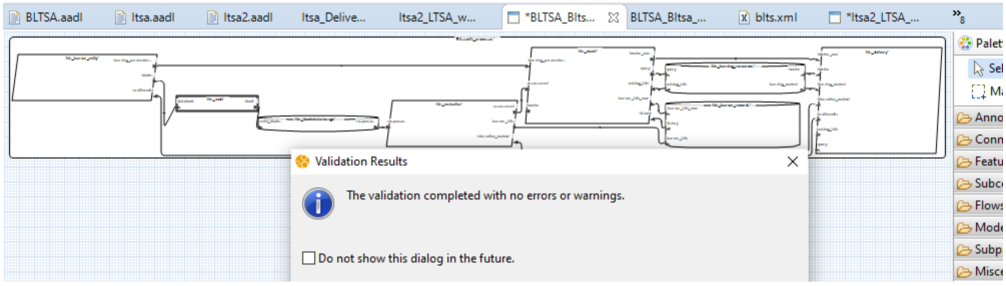


FIGURE 4.5: Validation of Blockchain Enabled Reference Architecture diagram in OSATE

The generated diagram is validated with the OSATE Tool. The Validation results are presented in Figure 4.5. The Validation results confirm that the Reference architecture proposed is valid, and no errors are reported.

Thus the proposed Reference Architecture's validation is completed. Now, a Scenario-based verification is computed in the next section.

4.3 Scenario based Verification of Blockchain enabled Smart Learning Environment Reference Architecture

As mentioned in the 3.2, a scenario that involves newly added/modified components is to be considered. We have chosen the Examination activity in the online learning environment. The major challenge in e-Learning is online examinations and assessments, as the evaluation results have a direct impact on the learning outcomes [57][8]. The Smart Learning Environment should be able to provide more secure and reliable, and tamper-proof data generated during assessments. IEEE LTSA has a centralized database. A centralized

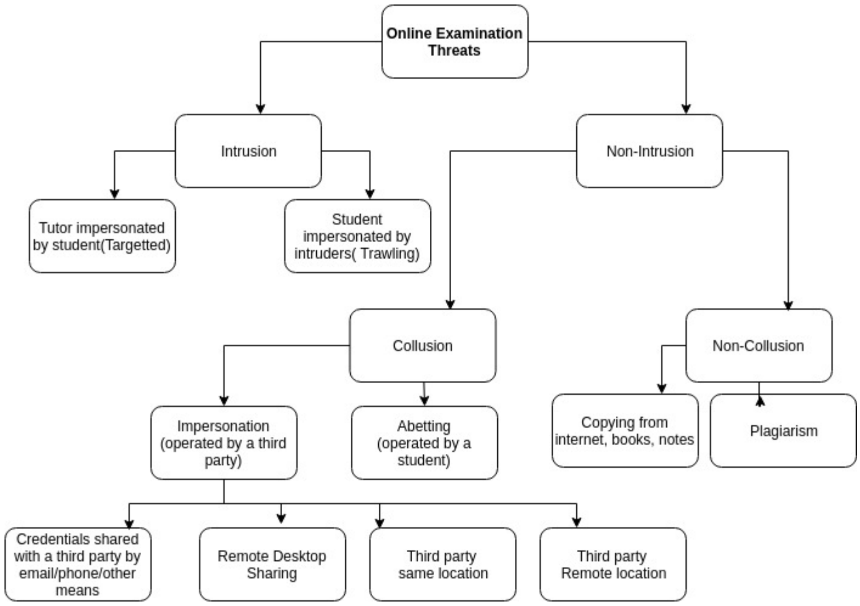


FIGURE 4.6: Threats Classification in remote Online examinations

database is always challenged against quality parameter availability. Any web-based centralized database is prone to several security breaches, and Data security is a global crisis. Axel[8] explained that major centralized systems are hacked multiple times a year. Abrar et al. [3] has presented a taxonomy of online examination threats presented in Figure 4.6.

For the proposed Reference architecture, the data to be stored in a block is prescribed as below. It is highly essential to record and analyze the behavioral data of the student during their interaction with the learning management systems. According to Kun Lian et al. [35], behavioral data helps to perform student profile analysis with Student Behavior Feature similarity calculation, Learning attitude analysis, Duration analysis of Online learning behavior. The proposed framework captures the behavioral analysis of a user. The difference in the behavior among the students will derive the learning patterns. It enables to predict the students' results by matching with the previous learner's behavioral data. It leads towards more personalization [29] of an online learning environment.

There is a need to secure the online data so that it becomes tamper-proof. Such sensitive data can be stored as blocks in the modified LTSA framework.

In the proposed framework, blocks store the data of a session which includes:

- User details

- The IP address of the system
- Responses recorded by the student/participant
- Behavioral data (Keyboard inputs, mouse movements)
- Images of the student (random per session) through web camera (optional)

For every predefined time duration (session), a hash of a block is generated from each user and immediately communicated to the local network node for validation. For a time slot t_i, with 'k' users, there would be $k * t_i$ number of blocks added to the blockchain network. The duration set T is a set of time slots of a user.

$$T = \{t_i | t_1, t_2, ...t_n\}$$

For example, if there is an examination with duration 60 minutes and a session is considered as 5 minutes each, then there would be 12 such sessions with 12(duration set) blocks being uploaded to blockchain network for a particular user.

The internal organization of blocks is viewed as a merkle tree [45]. A merkle tree consists of hash value of all transactions in a block. Each block has the hash value of previous block. New block is added to the tail block by placing the hash value of new block.

4.3.1 Algorithmic representation of the Work flow for Examination Scenario

The workflow for the online Examination scenario is presented Algorithmically at 2.

Algorithm 2: Algorithm for the online examination Scenario

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 $t_c < t_f$ *&* $t_t \neq NULL$ **do**
  if (
    $(n(t_c)\%s \leftarrow 0)$
) **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

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, 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 in 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 any local node generates a block, 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 the student submits the exam before the final time, a final block needs to be generated. The final block consists of the exam log 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 Scenario design, once the exam is done, the Evaluation process commences. The evaluation process can be carried by confirming the last block hash value with the Quiz (exam) responses hash value.

4.3.2 Activity Diagram

The activity diagram developed for the Online examination scenario's workflow is depicted in Figure 4.7

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 the exam in between, then the session will be closed after generating the final block. There are four 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.

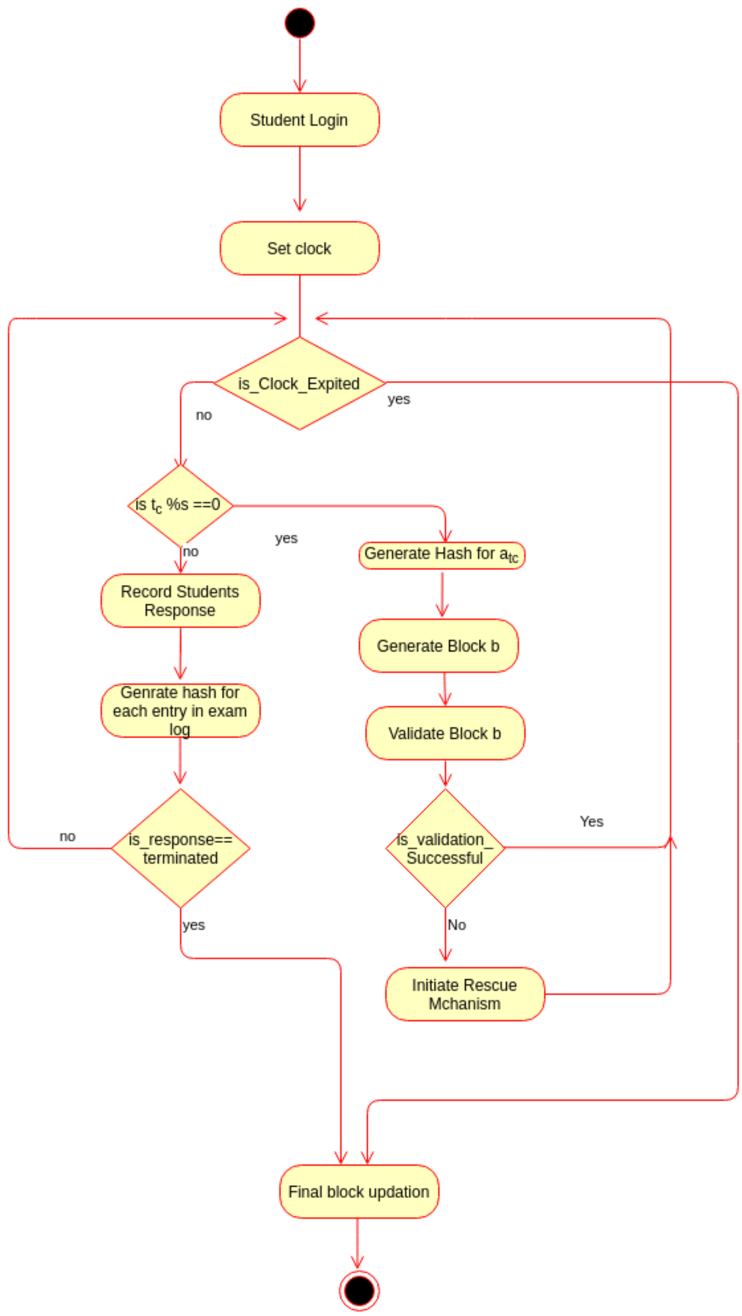


FIGURE 4.7: Activity diagram for the Online examination scenario

4.3.3 Petri Net Model for the Examination Scenario

A Petri net model, generated for the activity diagram of the examination scenario, is depicted in Figure 4.7. The procedure described [53] for converting the activity diagram into the Petri net is adopted and realized through the PIPE tool[55][52]. This scenario is for storing the response for a valid authentication. So, only a successful login case is considered. In the conversion process from workflow to a Petri net, the initial node is first taken as a place, i.e., "Start." The next Place will be " Student login," obtained by the transition after students provide their login credentials. Once this transition is fired, the student moves from the Start state to the "Login Successful" state. Similarly, the intermediary steps/actions required to get into the next state are considered as transitions represented with the rectangular bars, and circles represent the states.All Places and Transitions are presented in Tables 4.1, 4.2.

The generated Petri net is depicted below 4.8

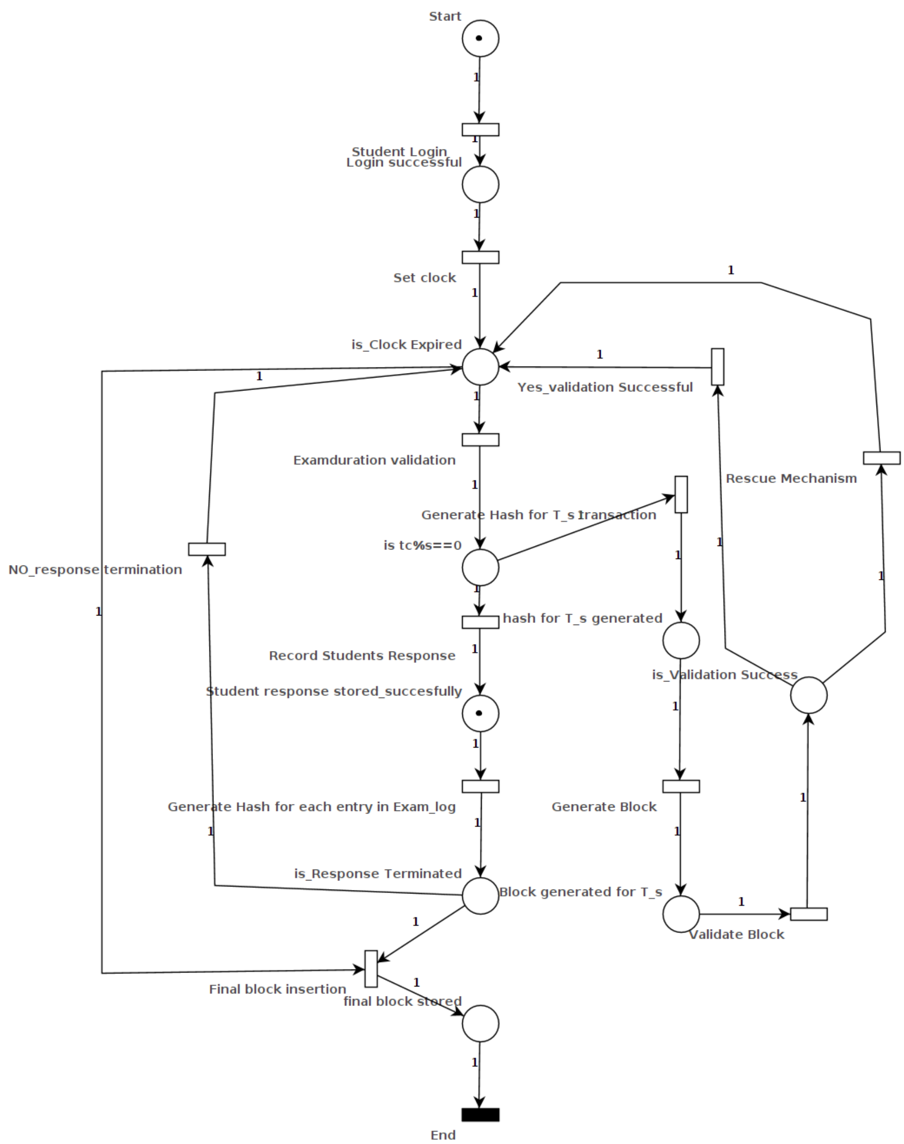


FIGURE 4.8: Workflow for verifying Requirements of SLE with Petri net formalism

4.3.4 Results & Discussion

The Generated Petri net (Figure 4.8) is analyzed through PIPE tool. There are ten places and thirteen Transitions in this 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
P7	Block generated for t_s
P8	is_Validation Success
P9	final block stored

Table 4.1: The Places 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

Table 4.2: The Transitions in the generated Petri net

4.3.5 Matrix Analysis

The Incidence matrix is an alternative way of describing a graph from conventional methods. The Incidence matrix D is obtained with the Backward and forward incidence matrices. The Backward incidence matrix is represented by D^- and the 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 the output matrix consisting of the elements like d_{ij}^+ which depicts the count of arcs joining transition t_i with place p_j. The matrix 4.2 represent the forward incidence matrix, 4.1 represent backward incidence matrix, 4.3 represent combined incidence matrix.

$$\begin{array}{c c}
 & \begin{array}{ccccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 & T11 & T12 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\end{array}$$

Matrix 4.1: Backward Incidence Matrix D^+

$$\begin{array}{c c}
 & \begin{array}{ccccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 & T11 & T12 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \end{array} &
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\end{array}$$

Matrix 4.2: Forward Incidence Matrix D^-

The combined incidence matrix D is defined as $D = D^+ - D^-$.

The incidence matrix represents the relation between places and Transitions. The generated incidence matrices are helpful in the analysis of the Petri net through matrix computations. The generation of the Reachability graph and further analysis is explained 4.3.6.

4.3.6 The Reachability Graph Analysis

The Petri net analysis begins with the generation of its state space and representing it with a Reachability Graph [2]. 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

$$
\begin{array}{cc}
 & \begin{array}{ccccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 & T11 & T12 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \\ P8 \\ P9 \end{array} &
\begin{pmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & -1 & 0 \\
0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1
\end{pmatrix}
\end{array}
$$

MATRIX 4.3: The Incidence matrix

model. The sequence of firing derives 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.

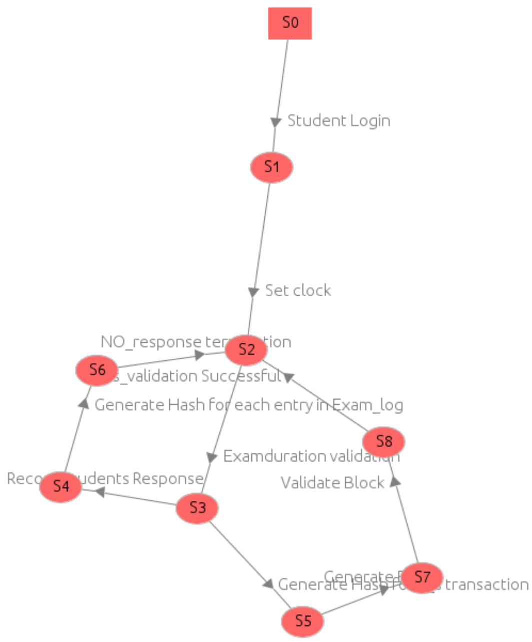


FIGURE 4.9: Reachability Graph R for the Petri net N

There are 9 distinct states in the generated Reachability graph. 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. Each state is a marking. The nine states are represented as nine markings in Matrix 4.6.

4.3.7 GSPN Analysis

The Stochastic Petri net becomes Markovian as it has both timed and immediate transitions. In a Markov process [58] 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.

4.3.7.1 Steady-State Analysis

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

$$\pi Q = 0 \tag{4.4}$$

$$\sum_{(i=1)}^{n} \pi_j = 1 \tag{4.5}$$

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. Matrix 4.6 represents the GSPN Steady-State Analysis Results matrix.

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

Matrix 4.6: GSPN Steady-State Analysis Results

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

TABLE 4.3: Steady-State Distribution of Tangible States

The Reachability graph has nine markings, out of which M0 is the initial marking which is represented with a square as depicted in Figure 4.9. 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.

4.3.7.2 Steady-State Distribution of Tangible States

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

$$E[T] = \frac{E[N]}{E[S]} \tag{4.7}$$

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.

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

4.3.7.3 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(p_i,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.

$$H(p_i, n) = M\epsilon R(PN)|M(p_i) = n \tag{4.8}$$

Transition	Throughput
Block generated for t_s	0.13793
Hash for Ts generated	0.13793
is_Clock Expired	0.27586
Is Response Terminated	0.13793
is_Validation Success	0.03448
Is $t_c\%s == 0$	0.13793
Login Success	0
Start	0
Student Response stored successfully	0.13793
final block stored	0

Table 4.4: Average Number of Tokens at a Place

A place p_i has an average count of tokens m_i attained by

$$\overline{m_i} = \sum_{n=1}^{\infty} nP[H(p_i, n)] \tag{4.9}$$

The table 4.4 represents the 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.

4.3.7.4 Probability of a Transition being Fired

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

$$k_j = \sum_{M_i \epsilon HN_j} \pi_i \left(\frac{\lambda_j}{-q_{ij}}\right) \tag{4.10}$$

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

Transition	**Throughput**
Exam duration 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
$Yes\_validationSuccessful$	0.03448
Final Block Insertion	0

TABLE 4.5: The throughput of Timed Transitions

4.3.7.5 Throughput of a Timed Transition

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

$$d_j = \sum_{s_i \epsilon HN_j} \pi_i \tag{4.11}$$

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 4.5.

4.3.8 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.

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.

Place	Average number of tokens	95% confidence interval(+/−)
Block generated for t_s	0.11881	0.06899
Hash for Ts generated	0.12871	0.07219
is_Clock Expired	0.21782	0.02263
Is Response Terminated	0.08911	0.08556
is_Validation Success	0.11881	0.06899
Is $t_c\%s == 0$	0.21782	0.01552
Login Success	0.0099	0
Start	0.0099	0
Student Response stored successfully	0.08911	0.08713
final block stored	0	0

TABLE 4.6: Petri nets Simulation Results

4.3.8.1 State Space Analysis Results

The State space analysis determines the qualitative properties of Petri net.

Petri net state space analysis results

Bounded	true
Safe	true
Deadlock	false

FIGURE 4.10: Petri nets Analysis Results

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

In addition to the above Architectural validation and Scenario based verification, we have also performed the Quantitative evaluation for Security with OSWAP's Risk Rating Methodology too. It is an attempt to measure the enhancement in security quantitatively.

4.4 Quantitative evaluation of security in Learning Management Systems

The proposed Reference Architecture is evaluated using a well-known risk analysis model for application viz., OWASP's Risk Rating Methodology [49] [6] . Scenarios can be developed to check the security quantitatively [4] [57] [3] The evaluation procedure is explained

Confidentiality of Impact (CI)	[9] All data disclosed [7] Critical data disclosed [6] Sensitive data disclosed [1] Non-Sensitive data disclosed
Integrity of Impact (II)	[9] All data totally corrupt [7] Extensive corrupt interrupt [6] Serious corrupt interrupt [1] Minimal corrupt interrupt
Availability of Impact (AI)	[9] All services cost [7] Extensive services interrupt [6] Serious services interrupt [1] Minimal service interrupt
Accountability of Impact (ACCI)	[9] Completely anonymous [7] Possibly traceable [1] Fully traceable

Table 4.7: Technical Impact Factors

as follows.

Let a set of scenarios be $S = \{s_1, s_2, s_3, ...s_n\}$

A scenario is defined as a tuple of requirement, threats, patterns i.e $s_i = (r_i, t_i, p_i)$ is a tuple where $r_i \epsilon R, t_i \epsilon T$ and $p_i \epsilon P$ such that R, T and P represent security requirements, threats, and security patterns respectively.

A requirement is a specific security prerequisite that describes the required security property of the system. A threat is a description of the risk to the system in which it explicitly or implicitly violates the requirement. Patterns are the practices that must be incorporated to mitigate the corresponding threat and safeguard the required behavior.

In risk based evaluation, risk factor is computed as a function of Technical Impact (I) and Vulnerability Factors(VF). The technical impact factors and vulnerability factors and their levels of severity is presented in Table 4.7 and Table 4.8 respectively.

$$\text{The Technical impact (I)} = \frac{\text{Sum of the severity of the technical factors}}{\text{Total number of factors considered}}$$

$$Is_i(\text{ Technical impact of scenario 'i'}) = \frac{I(CI_i)+I(AI_i)+I(II_i)+I(ACCI_i)}{4}$$

$$\text{Vulnerability factor(VF)} = \frac{\text{Sum of severity of the vulnerability factors}}{\text{Total number of factors considered}}$$

Ease of Discovery (ED)	[9] automated tool available [7] easy [3] difficult [1] practically impossible
Ease of Exploit (EE)	[9] Automated tool available [5] Easy [3] Difficult [1] Theoretical
Publicity (Pub)	[9] Public Knowledge [7] obvious [3] Hidden [1] Unknown

Table 4.8: Vulnerability Factor

Vulnerability factor for a scenario s_i is

$$VFs_i = \frac{ED_i + EE_i + pub_i}{3}$$

The risk impact factor Rs_i for the scenario s_i follows the standard risk equation

$$Rs_i = Is_i * VFs_i$$

Ten scenarios for Smart learning security assessment derived from literature is applied to the proposed blockchain enabled smart learning environment framework. For each threat type the risk impact factor is computed. The possible risk impacts are compared with the traditional IEEE LTSA. The results are specified in Table 4.9.

It is clearly evident that at each scenario, the new framework proposed has a significant reduction in the risk levels. It is obtained due to the immutability nature of the sophisticated blockchain technology and the consistent challenges being faced by the centralized database of the LTSA. It is observed that threats from outsiders are very nominal as compared to threats from insiders.

Impersonation in LTSA has more risk because there are several automated tools [11]available to crack the password where as it is quite complicated to get permission in private blockchain API from the network manager . Modification of transactions is widely possible in the LTSA, with the available automated tools or by other means. But in the proposed

Scenario	Threat type	Pattern	LTSA			Blockchain enabled SLE framework		
			Technical impact factor	Vulnerability factor	Risk impact factor	Technical impact factor	Vulnerability factor	Risk impact factor
1	Interception	Tutor impersonated by an intruder(outsider)	5.4	3.6	19.44	1.25	1.67	2.075
2	Interception	Tutor impersonated by a student (insider)	5	6	30	1.25	1.66	2.075
3	Modification	Modification of responses by insider	5.25	4.33	23.7325	1.25	2.66	3.325
4	Modification	Modification of responses by an intruder	7.25	6.33	45.89	2.5	1.66	4.15
5	Fabrication	Adding a spurious transaction by insider	5.25	5.66	29.715	2.25	1.66	3.735
6	Fabrication	Adding a spurious transaction by outsider	6.25	3.66	22.875	1.25	1	1.25
7	Abetting	A third party from the same location trying to write on behalf of a student	3.25	3.66	11.895	1.25	2.33	2.915
8	Abetting	A third party from a remote location trying to write on behalf of a student	5	7	35	1.66	2.33	3.867
9	Interruption	Interruption of services by outsiders	6.5	4.33	28.145	1.25	2.33	2.915
10	Interruption	Interruption of services by insiders	6	3	18	1.25	1	1.25

TABLE 4.9: Risk levels of LTSA and Blockchain enables SLE

framework it is hardly possible. The entire Merkle tree needs to be modified to commit a single transaction alteration.

- Even adding a spurious transaction to the database also replicates the above hardness
- Interruption of services has more chances in the LTSA centralized database system, whereas blockchain is a distributed one, more availability is ensured.

The pictorial representation of the quantitative results in Table 4.9 is as in Fig. 4.11.

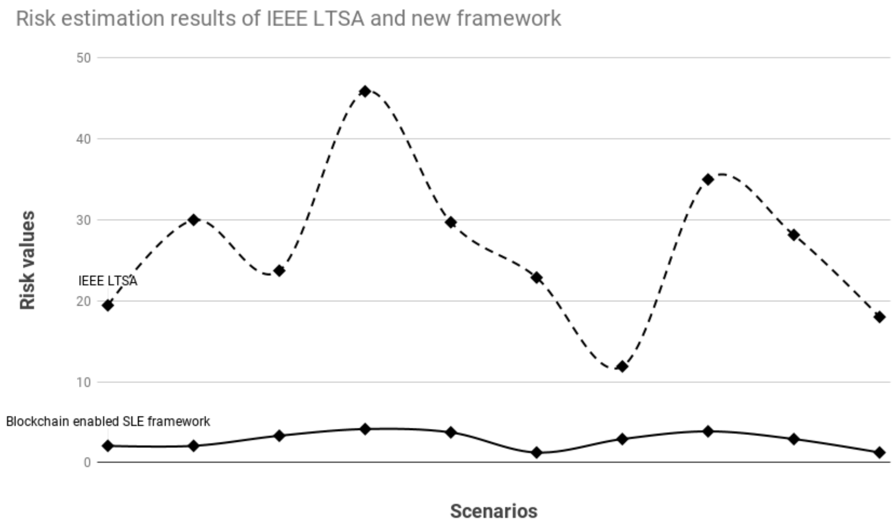


FIGURE 4.11: Risk factor comparison of LTSA and proposed architecture.

4.5 Conclusion

In this chapter, a Reference architecture for Blockchain-enabled Smart Learning Environments is proposed and validated. IEEE LTSA is extended to accommodate Blockchain's distributed structure. The proposed architecture is described in Architecture Analysis and Description Language(AADL). A Scenario is built for the most vulnerable learning activity "Examination", and formally verified with Petrinets.

Chapter 5

Design of a Middleware to address the Interoperability Standardization challenges.

Interoperability among e-learning content and system components, is a key to the successful implementation of an e-learning environment.

There are several e-learning interoperability specifications and standards at various stages of development and adoption that are being promoted by several organizations and consortiums.

Each interoperability standard has its advantages and disadvantages.

Systems Interoperability is a Standardization challenge in the first Smartness level. Peter Mikulecký [44], Yang et al. [66] described this challenge as an "interoperable operation between different standards of learning resources, services, and platforms." This standardization challenge addressed with an additional layer as a component is added to IEEE LTSA.

This additional layer enables the learning system for activity-based interoperability module selection. This chapter's organization is as follows: The Context and Motivation for this work are mentioned in the first section. The second section presents the proposed

Reference Architecture followed by Validation. The third section presents a Scenario-based verification with a Colored Petri net and a Discussion on Results. The final section gives the conclusion to this chapter.

5.1 Context and Motivation

Interoperability standards determine how e-Learning courses interacts with Learning Management Systems. The interoperability standards ensure that the content can be unified by systems, also creating a unified way for e-Learning systems for communication. The interoperability standards provide the feasibility to move the learning content among the learning systems. The learning data can be exchanged/relocated among the learning systems utilizing the same interoperability standard. This has limitations among the vendors(Learning services providers), customers [24]. The Interoperability standards are continuously evolving. Each standard has variety of defined features. As the standards are continually evolving, the modifiability of the existing systems become expensive. Thus, a strong component that supports multiple interoperability modules is proposed in this chapter.

Widely adopted Interoperable Standards are SCORM, AICC, xAPI, LTI, CM and etc. The interoperabiltiy standards are presented in Table 5.1.

SCORM[21] abbreviated as Shareable Content Object Reference Model, developed by Department of Defence, United States of America in 1997. SCORM is a suite of technical standards that enable web-based learning content and LMSs to communicate with each other [5]. The SCORM defines a Content Aggregation Model that allows e-learning contents to be put in a zip file which can then be loaded into any LMS. It basically integrates a set of standards, specifications and guidelines for creating interoperable reusable content. SCORM is widely used, there are multiple editions and versions(SCORM1, 1,1.1, 1.2, SCORM 2004 editions 1,2,3).

AICC is abbreviated as Aviation Industry CBT (Computer-Based Training) Committee[65]. It was initially developed for the aviation industry as a communication standard, later used in the education domain. It can be deployed on a remote server. AICC uses an

HTTP-based protocol(specifically “HTTP-based AICC//CMI Protocol”) for communication, known as HACP. In 1999, AICC was released with Javascript API as an alternative to HACP. This Javascript API was re-used in SCORM also with minor changes.

Experience API, or xAPI[47] [9], is a technical specification for tracking and sending learning statements to an LRS (Learning Record Store). This is a Representational State Transfer (REST) Web service based on JavaScript Object Notation (JSON) for its data format. xAPI is a specification that has produced an application program interface (API) that captures learning data in the stream and stores it in a repository as ”instruction” objects. xAPI can communicate between heterogeneous learning environments. This API captures data in a consistent format about a person or groups’ activities from many technologies. xAPI collects many features compared to other standards. These features help in rigorous analysis. Thus it is highly productive for the usage of Artificial Intelligence in the education domain.

CMI5 [18] is the latest specification for the current needs that utilized Experience API as a communication protocol and data model. It is produced by the AICC working group and launched by the xAPI community. The experts from the e-Learning interoperability standards developers like XAPI, AICC have jointly worked together to provide more feasibility than existing standards. In any of the above-discussed standards, the features/data type is pre-defined. i.e., cannot define a new data field/content type. Whereas in CMI5, designers of content can record virtually any data they need. CMI5 provides definition for necessary components for system interoperability such as packaging, launch, credential handshake, and consistent information model. CMI5 is released in 2016 for usage.

IMS Global [16] is an organization working on the formulation of future education and learning technology. IMS has produced an open standard LTI(Learning Tools Interoperability) [54] that permits Courseware and learning tools from various vendors to be launched from within a Learning system. LTI provides a common way to plug applications into LMS. LTI provides an easier, secure connection between learning applications and learning systems and tools.

Though there are many other interoperable standards, the above five are mostly used. The summery of the above standards are as follows:

	Release Date	Pages	Widely Used	Run-Time	Packaging	Metadata	Sequencing	Works Cross Domain
AICC HACP	Feb 1998	337	Yes	Yes	Yes	No	No	Yes
SCORM 1.0	Jan 2000	219	No	Yes	Yes	Yes	No	No
SCORM 1.1	Jan 2001	233	No	Yes	Yes	Yes	No	No
SCORM 1.2	Oct 2001	524	Yes	Yes	Yes	Yes	No	No
SCORM 2004 "1st Edition"	Jan 2004	1,027	No	Yes	Yes	Yes	Yes	No
SCORM 2004 2nd Edition	Jul 2004	1,219	Yes	Yes	Yes	Yes	Yes	No
SCORM 2004 3rd Edition	Oct 2006	1137	Yes	Yes	Yes	Yes	Yes	No
SCORM 2004 4th Edition	Mar 2009	1162	Yes	Yes	Yes	Yes	Yes	No
IMS Common Cartridge	Oct 2008	135	No	No	Yes	Yes	No	Yes
IMS LTI	May 2010	25	In Academic LMSs	Yes	No	No	No	Yes
The Experience API (xAPI)	April 26, 2013	85	Not Yet	Yes	Partial	No	No	Yes
cmi5 (a companion to xAPI)	June 1, 2016	48	Not Yet	Yes	Yes	No	No	Yes

FIGURE 5.1: Interoperability Standards comparison chart

- SCORM [21] : Widely used, there are multiple editions and versions(SCORM1, 1,1.1, 1.2, SCORM 2004 editions 1,2,3).
- AICC [65] : Very old, but can be deployed in separate server
- xAPI [47]: Advanced with several parameters, extensive analysis possible. Records any activity, flexible learning histories, all device support,
- LTI [54] : From IMS. One of the most secured, Plug-and -play service. Core and optional features.
- CMI5 [18] : Stands between SCORM, xAPI. Better packaging,even support same window launch mechanism, and etc

	AICC	SCORM 1.2	SCORM 2004	xAPI	cmi5
Course sequencing	N	N	Y	Y	Y
Completion, spent time, pass/fail tracking	N	N	Y	Y	Y
Advanced tracking (games, simulations, informal learning, offline learning, etc.)	N	N	N	Y	Y
Single scores	N	N	Y	Y	Y
Multiple scores	N	N	N	Y	Y
No web browser required	N	N	N	Y	Y
Mobile-friendliness	N	N	N	Y	Y

Figure 5.2: Feature comparison chart

The features comparison of some of the interoperability standards is mentioned in chart 5.1. Each has different combinations of features like working in the cross-domain, packaging facility, metadata, sequencing of learning activities, etc. The feature comparison is presented in figure 5.2. The collection of more features leads to better analysis. The higher Smartness levels (Levels 3,4,5 in 2.1 require machine intelligence. The advantage of more feature collection by an Interoperability standard helps attain machine intelligence-based predictions, recommendations, etc. But, it would cause a colossal data load on the Learner records. For every learning activity, a vigorous analysis may not be required. i.e., the importance and significance level of the activity varies. A simple standard with minimal features collection is sufficient in some cases.

It can be seen from 5.1 that there is a continuous evolution of interoperability standards in the literature. Once a learning management system is developed in compliance with a particular standard, it isn't easy to transfer to another standard. As discussed above, each standard has its advantage and disadvantage. When a new standard with sophisticated features, a more analytical capability is released, the quality parameters like Maintainability get affected. A Learning management system developed according to a single standard becomes a static Learning management system that may not support any other standards.

A Reference Architecture is mentioned in this chapter to address these limitations. An additional layer to IEEE LTSA is added with a pre-loaded Interoperable Engine. The significant advantages that can fetch from this system are independent development of the Interoperable Engine process. The interoperable engine's development becomes independent from the system, and the processes connected to this interoperable engine communicate through the APIs. This way of designing makes the system flexible for integrating new standards at this layer without affecting the remaining components of the system. As this is a middleware with simple connections(just with APIs) with other layers, the processes' dependability reduces. To address the further limitation of switching between the Interoperable Standards can be addressed through the proposed multi Interoperable Standard compliance system.

The proposed Reference architecture has a process in between the learner entity and the remaining system. It can be observed clearly in the layered structure of the LTSA.

5.2 Reference Architecture with Interoperability Engine

An additional layer is added to the basic IEEE LTSA framework. The layered structure of the IEEE LTSA is depicted in Figure 5.3.

5.2.1 Layered Architecture of IEEE LTSA

In the IEEE standard for LTSA, five refinement layers of architecture are specified, but only layer 3 (system components) is normative.

The layered structure is presented in the 5.3 .

This architecture applies to a broad range of learning scenarios. Description about each layer in the standard presented as:

- Learner and environment interactions: Concerns the learner's acquisition, transfer, exchange, formulation, discovery, etc., of knowledge and/or information through interaction with the environment.

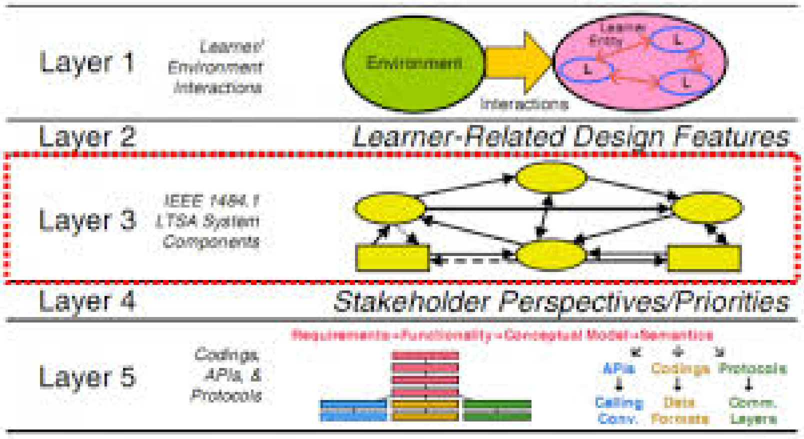


FIGURE 5.3: The layered structure of IEEE LTSA

- Learner-related design features: Concerns the effects learners have on the design of learning technology systems.
- System components (normative): Describes the component-based architecture identified in human-centered and pervasive features.
- Implementation perspectives and priorities: Describes learning technology systems from various perspectives by reference to subsets of the system components layer.
- Operational components and interoperability—codings, APIs, protocols: Describes the generic "plug-n-play" (interoperable) components and interfaces of an information technology-based learning technology architecture, as identified in the stakeholder perspectives.

The same can be represented as high level technical architecture as The Presentation layer has interaction from the Business layer, i.e., with Delivery and Evaluation.

The Delivery process interacts with the diverse environment of the Learning Entity. Interoperability standard defines the pre-defined instructions/protocol for communication between these two layers. The Delivery must be loaded with the complete instruction set that supports various learning activities and learning devices.

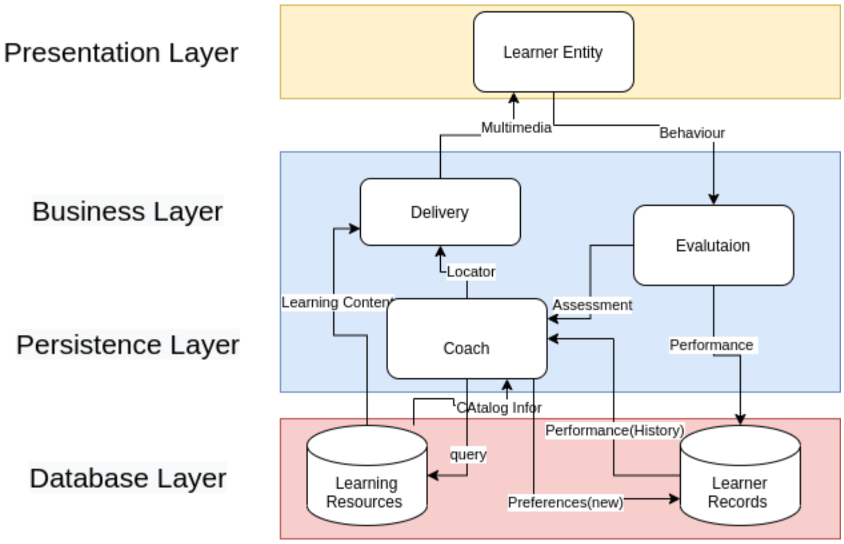


FIGURE 5.4: The IEEE LTSA system components in the layered structure

5.2.2 Reference Architecture with Interoperability Engine

The high-level Reference architecture with Interoperability Engine as a middleware to IEEE LTSA is depicted in Figure 5.5.

This Reference architecture is conformance to IEEE LTSA. As per the 2.3.3.1, all the processes of IEEE LTSA present in this architecture. A new process, "Interoperable Engine," is introduced to the learning system. The Delivery process and Interoperable Engine should interact through function calls with few arguments. The interoperable Engine process maintains multiple Interoperable standards. It selects an appropriate module based on the Delivery process request call. The content creators must select the type of analysis/features they anticipate from the learning activity. The function call from Delivery to the Interoperable Engine contains the arguments such as Learner details, learning activity, the interoperability module, etc. The Interoperable Engine calls the specific module that proceeds with interaction with the learner. This design facilitates multiple interoperable modules being part of the learning system. The modularity achieved with this approach is discussed in the layered structure presentation.

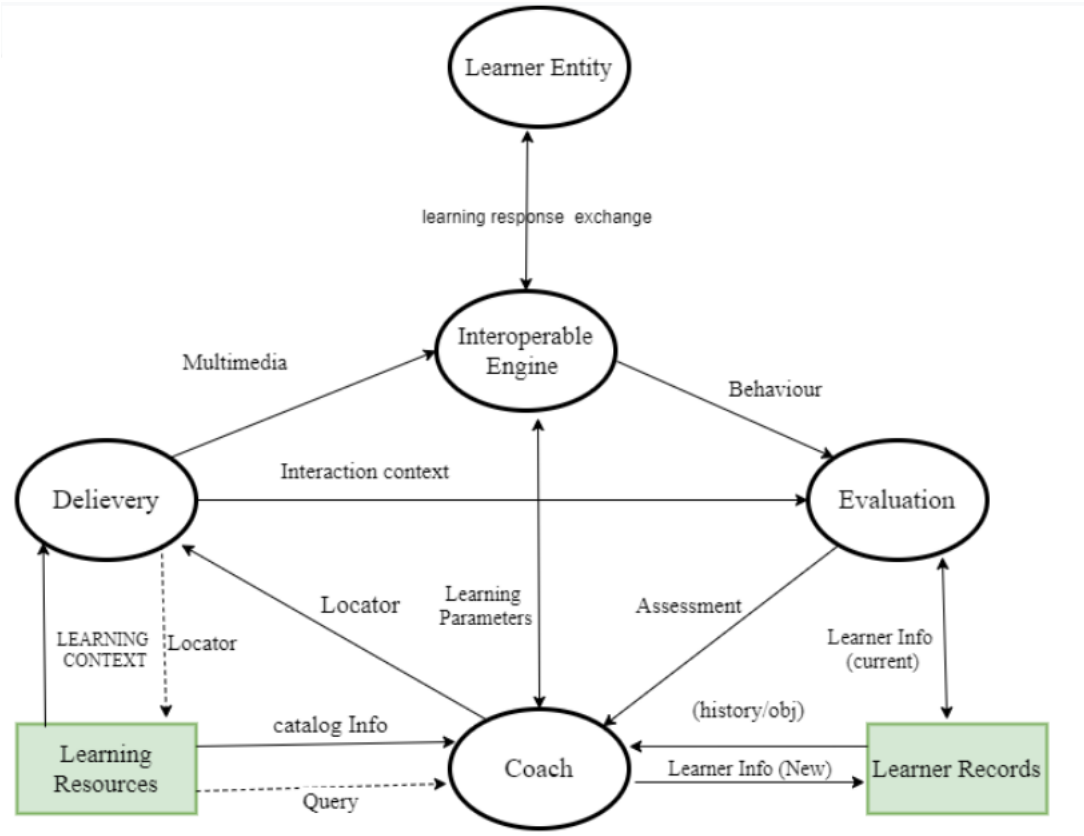


FIGURE 5.5: Reference Architecture with Interoperable Engine

5.2.3 Extended Reference Architecture with Interoperable Engine(Layered form)

”Interoperable Engine” is the additional layer incorporated to the IEEE standard for LTSA.

The interactions between the Business and Presentation layers are maintained the same, but Interoperable Engine as a middleware. The general interaction between these two layers is in three ways. One is ”Multimedia” that transmits from the Deliver process to the Learning Entity, and the other is ”Behaviour” that transfers from the Learning entity to the Evaluation. The ”Learning parameters” are exchanged between Coach to Learning Entity.

The new Interoperable Engine process acts as a middleware between the Business layer and Presentation layer. This middleware is like a hyper-visor in the virtualization of an Operating System. As discussed above, the Delivery process transfers the activity details to the interoperable engine. Once the activity details and student details are transferred to the Interoperable Engine, it will initiate a call to the particular module. Once a call is received, the respective module will start engaging with the Learning entity.

For example, if there is a function call for a Quiz(exam) with the SCORM module, the Interoperable Engine activates the SCORM module. This module gets the Quiz data, Learner details, and the learning activity is executed. The SCORM module manages the behavioral data collection from the Learner entity. As per Feature comparison chart5.2, each Interoperability standard collects various features, the SCORM module collects Final Score, Quiz completion status, the time spent on the Quiz, and finally, the pass/fail tracking. The collected data get to share with the Evaluation process in the lower layer.

The Evaluation process transfers that performance to Learner Records and gets stored here. The major advantage obtained is we can maintain the interoperable standard modules isolated, and even they can be deployed in different servers. Thus, the system will be open to the many modules, and their replacement or modifications does not affect the remaining learning system. Each interoperability module has its specifications, implementation details, etc., and their execution is mutually exclusive for a single activity. While content is being prepared, the content creator has to select an appropriate interoperability module with it. The same will be executed for any learner attempting that contents activity.

The proposed Reference architecture with new layer is depicted in Figure 5.6.

5.2.4 Proposed Reference Architecture Validation

The proposed Reference Architecture is described with AADL architecture description language on OSATE tool. The fragment of the code is presented at 5.7

All IEEE LTSA components(processes, interactions) are described in this Reference Architecture. An additional component, "Interoperable Engine," is also defined along with the interactions mentioned in the 5.5.

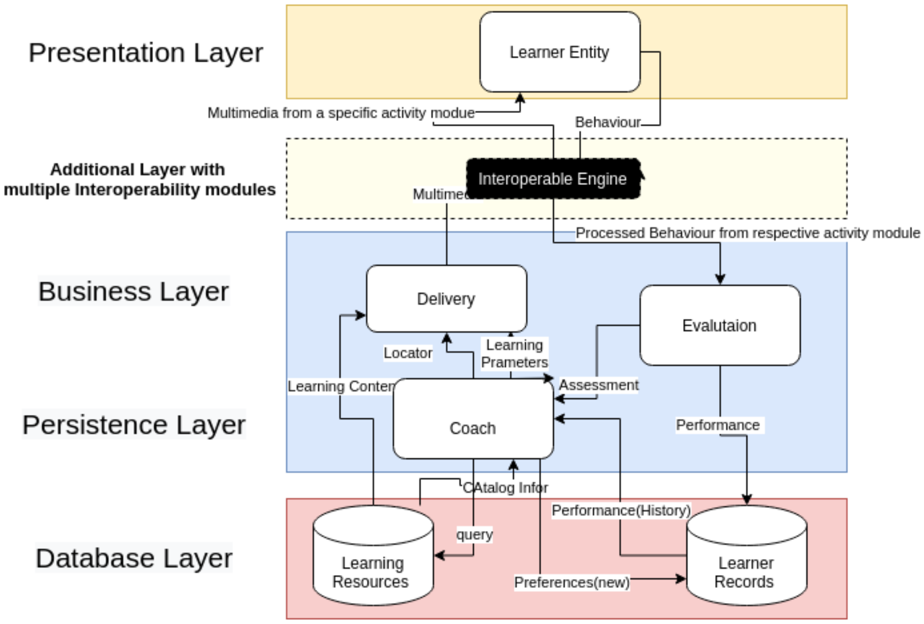


FIGURE 5.6: LTSA with Interoperable engine as additional layer

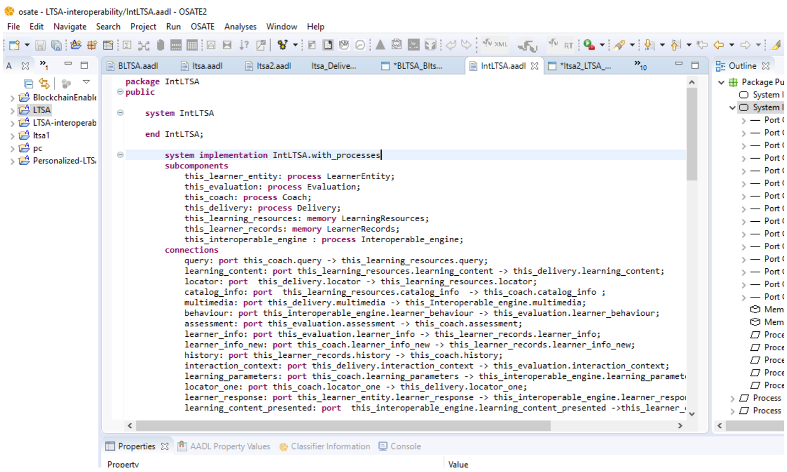


FIGURE 5.7: Reference Architecture implementation in OSATE with Interoperable Engine

5.2.5 Generation of Architecture diagram from ADL

The OSATE tool can generate an architecture diagram from the code described in Architecture analysis and Description Language. The architecture diagram generated from the Reference Architectures' code is presented in Figure5.8.

It can be observed that all the processes, memory, interactions are integrated properly.

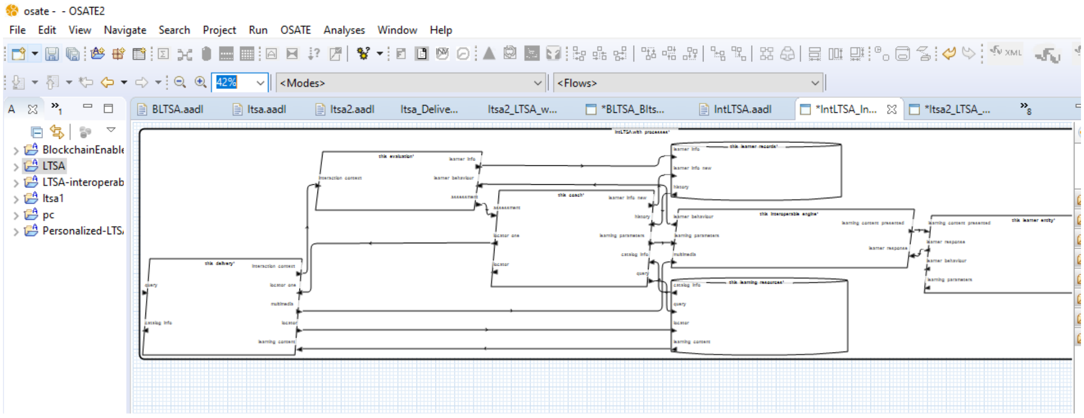


FIGURE 5.8: Generated Reference Architecture with Interoperable Engine

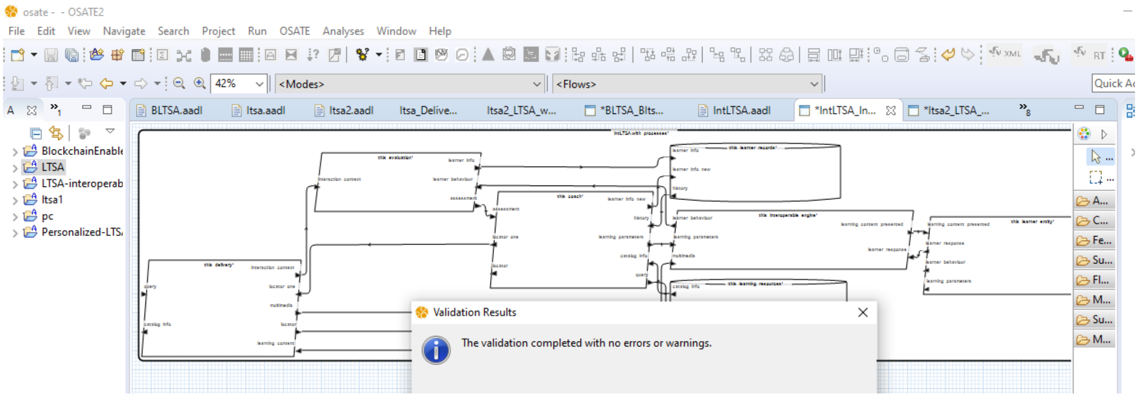


FIGURE 5.9: Validation of Reference Architecture diagram with Interoperable Engine in OSATE

The generated diagram is validated with the OSATE Tool. The Validation results are presented in the diagram 5.9. The Validation results confirmed that the Reference architecture proposed is valid and no errors are reported.

Thus the proposed Reference Architecture's validation is completed. Now, a Scenario-based verification is computed in the next section.

5.3 Scenario-based Verification of Smart Learning Environment Reference Architecture with Interoperable Engine middleware

5.3.1 Scenario based Verification of LTSA with Interoperable Engine

A Scenario with "Activity-based module selection" is designed to verify the newly added layer to LTSA. Interoperability standard modules in this layer are independent of each other. They can even be deployed on separate servers too. For any single learning activity, they are mutually exclusive. An activity designed with a particular interoperable standard does not associate with any other standards' module. There is mutual exclusion among modules of the standards while executing any learning activity.

In this scenario, the middleware considers five Interoperable standards, namely SCORM, xAPI, CMI5, AICC, LTI. There can be many such modules in this middleware; the flow between the Business and Presentation layers remains the same. This middleware acts as a connector between these two layers. Colored Petri nets can better perform the visualization of mutual exclusiveness in a workflow. The colored Petri nets have an extra variable set called "Colors," along with the five tuples of a Simple Petri net as mentioned in the Definition of a Petri net in Section 3.2.1. Firing a token from one Place to a Transition occurs by considering the weight of the token and the mandated color of the token.

This Scenario's workflow is as follows: Preconditions: Course Activities are designed along with the Pre-selected interoperability module. Interoperability Engine is loaded with all the modules that are being made available. Let 'A' be the set of activities, 'M' be the set of modules, *"id"* is the learner id.

$$\text{Activities A}=a_1,\ a_2,...a_n$$
$$\text{Modules M}=\ m_1,\ m_2,...m_n$$

Workflow:

1. An activity a_i is activated by sending a call with $(a_i,m_i,id) \ni a_i \rightarrow$ A, $m_i \rightarrow M$

2. Interoperable Engine calls the specific module mentioned in the above call.

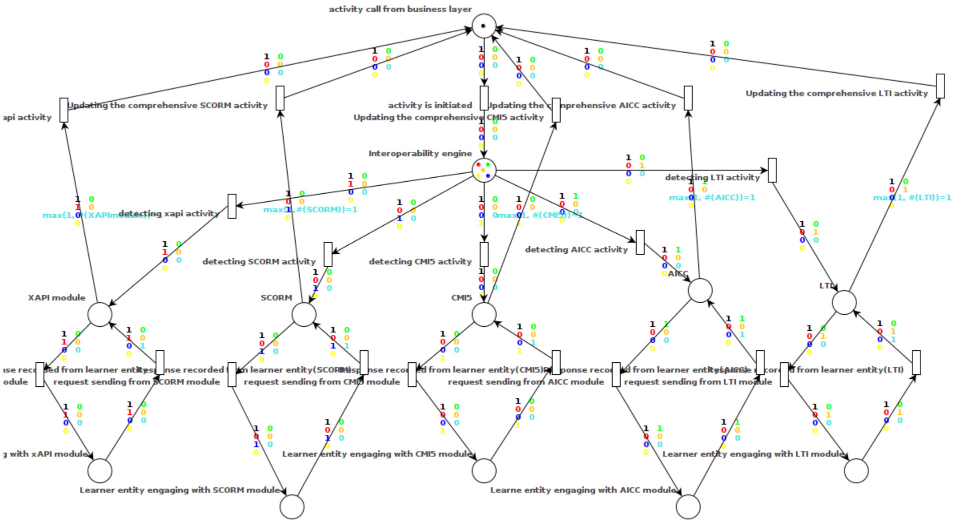


FIGURE 5.10: Petrinet for the Interoperable Engine

3. The module being called will initiate session with the learner entity *id*
4. The activated module collects the *behaviour* of the learner
5. The collected *behaviour* is returned to the lower level components.(*Evaluation*)

5.3.2 Petrinet

Petrinet for the above Workflow is generated from the PIPE[52] tool. Colored petrinets are used in order to facilitate the mutually exclusive selection of interoperability modules. The Generated petrinet is presented in Figure 5.10.

Note: The Colored Petri nets do not have a Reachability graph and CGPN analysis as the Reachability graph nodes deal just with the token weight but not with colors. By considering all tokens in a single color can generate a Reachability graph. But, it does have Behavioral properties.

The Places in the Petrinet are presented in Table 5.1.

The transitions in the Petrinet 5.10 are presented in Table 5.2.

Place	Place Name
P0	Interoperability engine_Default
P1	Activity call from business layer_Default
P2	AICC_Default_AICC
P3	Interoperability engine_Default_AICC
P4	CMI5_Default_CMI5
P5	Interoperability engine_Default_CMI5
P6	LTI_Default_LTI
P7	Interoperability engine_Default_LTI
P8	SCORM_Default_SCORM
P9	Interoperability engine_Default_SCORM
P10	XAPI module _Default_XAPI
P11	Interoperability engine_Default_XAPI
P12	Learner entity engaging with AICC module_Default_AICC
P13	Learner entity engaging with CMI5 module_Default_CMI5
P14	Learner entity engaging with LTI module_Default_LTI
P15	Learner entity engaging with SCORM module_Default_SCORM
P16	Learning Entity engaging with xAPI module_Default_XAPI
P17	XAPI module _Default_XAPI_Conformance
P18	AICC_Default_AICC_Conformance
P19	CMI5_Default_CMI5_Conformance
P20	LTI_Default_LTI_Conformance
P21	SCORM_Default_SCORM_Conformance

Table 5.1: Places list of the generated Petri net

5.4 Petrinet Simulation and Results

The Simulation Results are presented in Figure 5.11. The simulation is performed with 100 firings and 5 replications. For a petrinet with no deadlocks, the simulation includes all places and the values being derived will be based on the random firing. The presence of all places is the point need to be observed. The Average number of tokens

Transition	**Transition name**
T0	activity is initiated
T1	detecting AICC activity
T2	detecting CMI5 activity
T4	detecting LTI activity
T5	detecting SCORM activity
T6	detecting xapi activity
T7	request sending from AICC module
T8	request sending from CMI5 module
T9	request sending from LTI module
T10	request sending from SCORM module
T11	request sending from xapi module
T12	Response recorded from learner entity
T13	Response recorded from learner entity(AICC)
T14	Response recorded from learner entity(CMI5)
T15	Response recorded from learner entity(LTI)
T16	Response recorded from learner entity(SCORM)
T17	Updating the comprehensive AICC activity
T18	Updating the comprehensive CMI5 activity
T19	Updating the comprehensive LTI activity
T20	Updating the comprehensive SCORM activity
T21	Updating the comprehensive xapi activity

TABLE 5.2: Transitions list of the generated Petri net

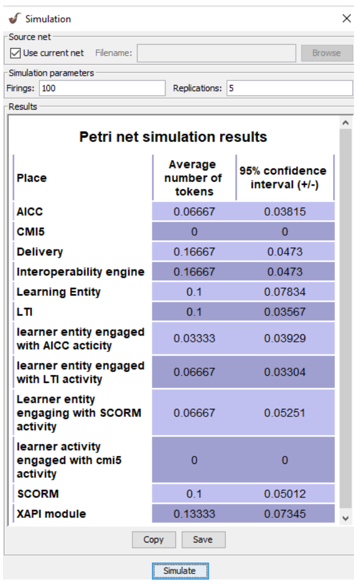


FIGURE 5.11: Simulation Results for Interoperable Engine

5.4.1 State Space Analysis Results

The state space analysis results presents the behavioral properties of the Petrinet. The results are depicted in Figure 5.12.

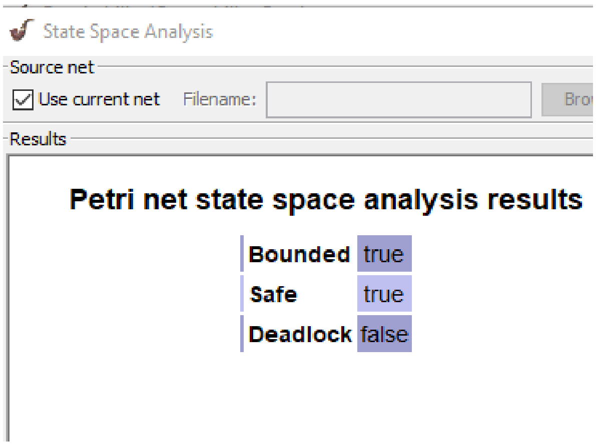


FIGURE 5.12: State space Results for Interoperable engine Petri net

It is evident that there are no Deadlocks, and it is Safe and Bounded. It confirms the Verification of the the generated Scenario and its Petrinet.

5.5 Conclusion

The proposed Reference Architecture with the new middleware layer is Validated successfully with OSATE. The Behavioral properties results 5.12 confirms the Petri net is Bounded, Safe, and Deadlock free. These results complete the verification of the generated Petri net successfully. Thus the scenario is formally verified. Therefore the proposed Reference Architecture can be considered for further development for a multi interoperability module-based learning system.

Chapter 6

Design of an Intelligent Coach to realize Personalized instruction delivery

The Smartness levels move from Human intelligence towards Machine driven intelligence as defined in the Smartness levels 2.1. The third to sixth levels of Smartness, i.e., *Infer, Learn, Anticipate, Self-organize* is associated with Artificial Intelligence. This chapter attempts to propose a Reference architecture that facilitates Personalization as a component in the learning system. The Pipe-Filter reference model is adopted to IEEE LTSA by incorporating an intelligent component as a filter.

6.1 Context and Motivation

The word "Smart" itself highlights the learning system's ability to be intelligent enough to address Context-aware, Ubiquitous learning assistance, Analytical capability to achieve a higher success rate in learning outcomes. To generate Personalized delivery instructions, we require Artificial intelligence assistance. Personalization is an emerging pedagogy in Smart Learning Environments[50].

It is an attempt to achieve Personalization as a Service in the learning system through the Intelligent component as a filter that provides this service. In general, the process "Coach" gets the "Learner info" from the "Learner Records" and "catalog" from the "Learning Resources." The Coach process returns the locator of the content and Learner_id to the "Delivery" process. It is a static procedure in IEEE LTSA for instructional delivery, which means the same instruction for every Learner. It is necessary to provide personalized instructions in a Smart Learning environment. In a way, intelligence modules can be placed as a part of "Coach," but maintainability gets affected. The process "Coach" has high interactions with all other processes. Any modification/change in this process will affect all other processes in this system. Also, the recommendations, prediction modules get updated frequently. The personalized services' implementations can be changed with the evolution of better algorithms. Thus separating these modules can maintain better modularization of the system.

6.2 Intelligent Coach to realize personalized instruction delivery in Smart Learning Environment

"Pipe-Filter" software architecture pattern is adapted to IEEE LTSA for Smart Learning Environments' personalization context. The proposed intelligent component of the learning system works as a filter for instructional delivery. The filter is like an additional component to the system. Its implementation can be done independently. A filter can be replaced with another with modifying interactions, which ensures loose coupling. The filter can even be called through Software as a service or with function calls. The computation within the filter does not affect the remaining system. In a pipe and filter architecture, the Filter is responsible for a particular task(personalization in this context). This Filter provides the Re-usability of its component. Thus, we can adapt it to any other learning system.

The newly added component(as a filter) does get the query from the Coach for a personalization service. Based on the Knowledge discovery algorithms, user representation models, E-Learning models/ontology developed in this filter, a recommendation is created.

The recommendation derived should be communicated to the coach. Based on the recommendation, the Coach selects the appropriate level of content and delivers the instruction. With this intelligent component filter, the Coach can be called a smart Coach.

The proposed Reference Architecture is presented in Figure 6.1.

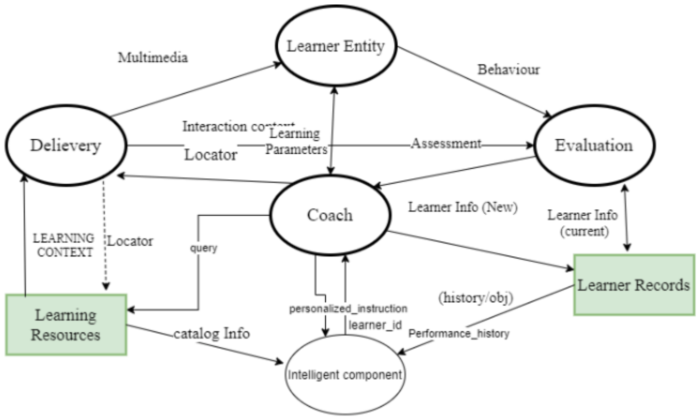


FIGURE 6.1: Reference Architecture with Intelligent component

All the IEEE LTSA components are present in the proposed Reference Architecture; thus, it is conformance to the IEEE LTSA. A new component with the name "Intelligent component" is added to the IEEE LTSA. It has a bidirectional interaction with the Coach process. A query along with Learner_id is the data transferred from the Coach process to the new component. Based on the learner_id, it gets catalog_info from the Learning Resources and performance_history from Learner Records. The intelligent component executes the query with the additional data derived from the other components and sends the personalized_instruction to the Coach.

6.2.1 Proposed Reference Architectures' Validation

The proposed Reference Architecture with Intelligent component is described with AADL architecture description language on OSATE tool. The fragment of the code is presented in Figure 6.2

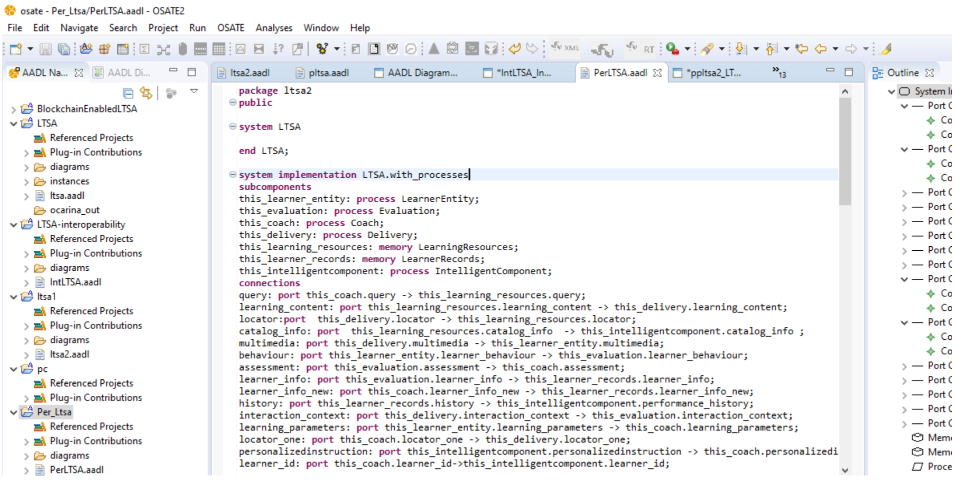


FIGURE 6.2: Reference Architecture implemented in OSATE with Intelligent component

All the processes, memories, and connections of IEEE LTSA are present in the code. A new component with the name "intelligent component" is created as a process. The new interactions are present in the connections of the Reference Architecture.

6.2.2 Generation of Architecture diagram from ADL

The Reference Architecture diagram with OSATE tool is generated from the code above in AADL. The generated diagram is presented at 6.3

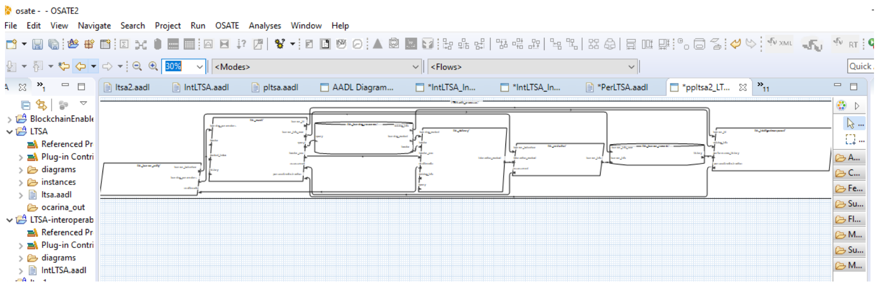


FIGURE 6.3: Generated Reference Architecture with Intelligent component

It can be observed that all the processes, memory, interactions are integrated properly. The newly added "Intelligent component" is added as a process to the learning system.

The generated diagram is validated with the OSATE Tool. The Validation results are presented in Figure 6.4. The Validation results confirm that the Reference architecture proposed is valid and no errors are reported.

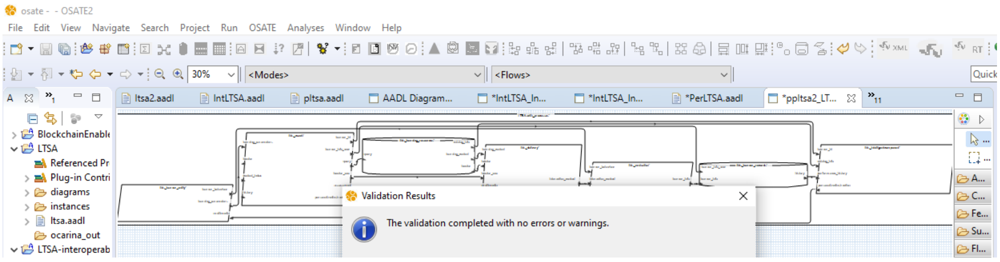


FIGURE 6.4: Validation Result of the Reference Architecture implemented in OSATE with Intelligent component

Thus the proposed Reference Architecture's validation is completed. Now, a Scenario based verification is computed in the next section.

6.3 Scenario based verification of the proposed Reference Architecture with design of an intelligent coach

There are several personalization services like Personal assistance, deriving Learning patterns, discovering Learning paths, Learner performance predictions, content recommendations, etc. The proposed architecture is realized with a scenario that generates a Personalized Dynamic learning plan. This plan gets updated for each user dynamically based on the completion of the course activities. 'Dropout' is one of the significant constraints faced by the e-learning industry. In the process of reducing the dropout-rate, personalized assistance must be provided to the risk zone students [48]. There are various personalized methodologies for recommendation and prediction available in the literature. Some predicting methodologies proposed by C.Taylor et al. [15]. Ya-Han Hu et al. [28] assesses and predicts the performance of students with various machine learning techniques. However, these Prediction algorithms are limited to estimate the future performance [31] followed by reporting to teacher/manager. Further, they do not provide instructions for improvement to the student automatically.

Online courses have a specific course duration realized by the course beginning date and course completion date.

Especially paid courses, semester-wise courses, competitive exams training courses do have the last date to finish. Each online course contains a set of activities stored in several formats. Online courses activities include lecture videos, documents/notes, questionnaire/quizzes/tests, etc [20]. The learner should complete each activity within the deadline of the course. Each activity is assessed formatively and summatively. In such a scenario, we need a plan to fulfill all these activities. Online courses have course plans which are common for all students. Each Learning Management systems[15] supports the common static course plans i.e. a fixed course plan common for everyone. As mentioned by D.F.O. Onah et al. [28] when the dropout rate is high, a common plan is not sufficient as people delay in completing the activities most of the time. This constraint is the source of motivation for the design and implementation of a Personalized Dynamic Learning plan with timely instructions. The plan is adaptive, which aids in the successful completion of the course.

6.3.1 Workflow of Personalized Learning Plan Generator Scenario

The Dynamic Learning Plan (DLP) helps in adjusting the schedule, which is adaptive to the learner's progress. Any Smart learning system can adopt the proposed approach. The Methodology for dynamically updating and upgrading the plan is presented in Figure 6.5. Learning Resources and Learner Records are the data stores in the IEEE Standard for

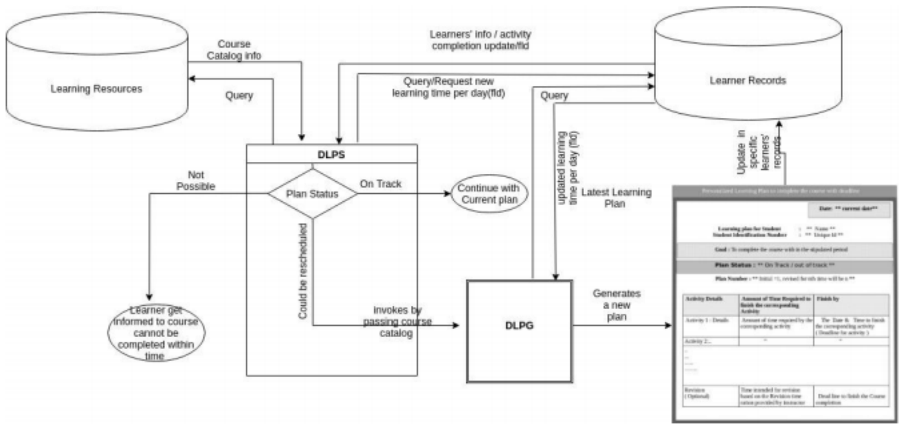


FIGURE 6.5: Workflow of Dynamic Learning Plan

Learning Management Systems Architecture [39]. Learning Records contains the stakeholders' profile details, courses enrolled, assessment reports, course activity tables, etc. Generally, a course plan is stored in the learning resources(If it is included in the system). It is proposed to include a new entity named learner plan, which has to be included in learning records to facilitate individual learning plan. This entry has its significance, as a static plan is common for a course, and an individual learning plan is specific for each learner. Dynamic Learning Plan Selector(DLPS) gets invoked whenever a learner's activity gets updated (viewed/completed/etc.). The initial course plan is stored as a "current plan" for each learner. There are three possible scenarios while generating a learning plan,

1. A learner following the "current plan" successfully

2. A learner is out of track/ not following the current schedule, but rather, there is a scope for completing the course with increased feasible learning time per day (new_fld)

3. A learner is out of the track, and there is no scope for completing the course, thus a learner is informed that there is no abundant amount of time available to complete the course.

DLPS addresses each scenario and proceeds with necessary actions. When the learner is following the current plan as per schedule, then no action is needed. Otherwise, the status is set to "Could be rescheduled," and the learner is informed that the feasible learning time per day (fld), i.e., the amount of time that can be spent per day needs to be readjusted. Based on the "new_fld" value given by the learner, it is decided whether the learner can still complete the course within time. If so, DLPG is invoked. If not, the status is set to "Not possible," and the learner is informed about it. Dynamic Learning Plan Generator(DLPG) generates a modified personalized learning plan, and the same is updated in the corresponding learner's record. A student gets a modified and updated plan with which the course can be completed within the available time.

6.3.2 Algorithm for Dynamic Learning Plan Generator

The Dynamic Learning Plan Generator (DLPG) algorithm:
The Dynamic Learning plan algorithm is adaptive in nature and it considers all possible scenarios. It consists of two routines namely: Dynamic learning plan selector (DLPS) and Dynamic Learning plan generator(DLPG). DLPS evaluates the current scenario through the Plan status which leads to the following possible cases.

Case 1:	**Plan status = On Track** *Action:*Current plan is continued
Case 2:	**Plan status = Could be rescheduled** *Action:* Learner is out of track with their learning plan. There is a possibility for the Learner to complete the course, if they can increase the amount of time that they can spend on learning per day. In such a case, a new plan is generated by invoking DLPG. This can be controlled by the LMS admin, by placing a threshold. This threshold can be decided based on average amount of time that one can spend on electronic devices and time available to complete the course
Case 3:	**Plan status = Not possible** *Action:* Available time duration may not be sufficient for completing the course. The user is informed about the same. Practically, this could save the Learners' effort and time, which they can use it for completing the course in next session.

TABLE 6.1: DLPG cases, conditions and actions

The data and algorithm of Dynamic Learning Plan Selector are as follows:

Algorithm 3: Dynamic learning plan selector

Data: Course details: Course Deadline 'DL' , Current Date 'CD', Total content hours 'TCH', Remaining Days 'RD', 'n' is the number of course activities, Remaining content hours 'RCH', Course Activity details C = $[a_1, a_2, a_3, ...a_{N+1}]$, time required for each activity T =$[t_1, t_2, t_3, ..t_{n+1}]$ where t_i is the time required to complete activity a_i, User 'u' details from learner's records like Activity Status, Plan status 'PS', Plan no 'P', Feasible learning time per day as 'fld'

Result: Decide the appropriate case and action for individual learning plan

if (

  $RD * fld \geq RCH$

) **then**

  Continue with the current plan, Plan status remains "On Track"

else

  Plan status ← " Could be rescheduled" ;

  new_fld ← read new fld from u ;

  if (

    *RD * new_fld* $\geq$ *RCH*

) **then**

    Plan status ← 'Could be rescheduled';

    $fld \leftarrow new\_fld$

    call DLPG();

  else

    Plan status ← 'Not Possible";

    Inform u that with new_fld course cannot be completed

  end

end

The DLPS algorithm takes the latest feasible learning time per day(new_fld) as input and checks the possibility of course completion. If a plan needs to be generated for the first time or updated, it invokes the Dynamic Learning plan Generator algorithm(DLPG)

The DLPG algorithm is as follows at Algorithm4:

Algorithm 4: Dynamic learning plan Generator

Data: Course details: Course Deadline 'DL' , Current Date 'CD', Total content hours 'TCH', Remaining Days 'RD', Remaining content hours 'RCH', 'n' is the number of course activities Course Activity details A=$[a_1, a_2, a_3, ...a_{n+1}]$ time required for each activity T =$[t_1, t_2, t_3, ..t_{n+1}]$ where t_i is the time required to complete activity a_i, User 'u' details from learner's records like Activity Status, Plan status 'PS', Plan no 'P', Feasible learning time per day as 'fld', per day scheduled hours 'PSH'

Result: Learning plan with scheduled duration for each activity

Update RCH by adding the duration of unexplored activities a_i in A

$P \leftarrow P + 1$

$RD \leftarrow DL - CD$

$PSH \leftarrow \frac{RCH}{RD}$

for $i \leftarrow 1$ **to** n **do**

  if (

    activity a_i is not completed in A

) **then**

    assign the 'x' number of days for that activity where

$$x = \lceil \frac{a_i}{RCH} * 100 \rceil \tag{6.1}$$

  end

end

Update the Learning plan parameters in Learner's records

Export the learner plan details to Learning plan Template and display

---

Result: Learning plan with scheduled duration for each activity .

The Individual learning plan is displayed to the user with the learning plan template. The learning plan template facilitate the required plan numbers and the regularly updated activities and corresponding time schedule for each pending activity.

6.3.3 Petrinet for the Scenario

The Petrinet for the above Dynamic learning plan Generator is depicted in Figure 6.6.

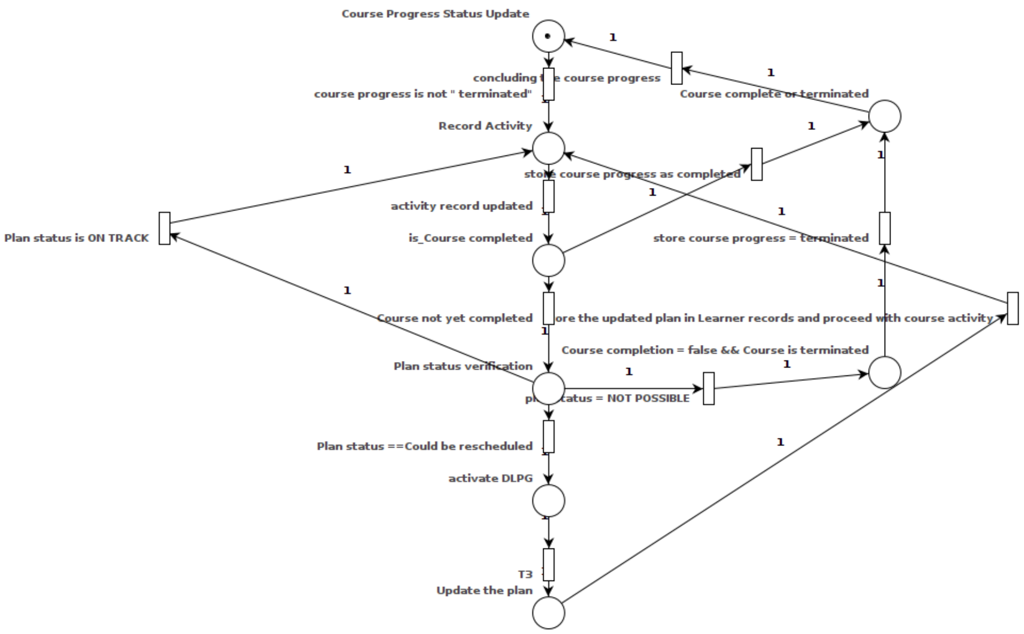


FIGURE 6.6: Petrinet for the Dynamic learning plan generator

There are 8 places and 11 transitions in the Petrinet. They are listed in Tables 6.2,6.3.

Place	Place Name
P0	activate DLPG
P1	Course complete or terminated
P2	Course completion = false && Course is terminated
P3	Course Progress Status Update
P4	is_Course completed
P5	Plan status verification
P6	Record Activity
P7	Update the plan

TABLE 6.2: The Places in the generated Petri net

Transition	**Transition Name**
T0	activity record updated
T1	concluding the course progress
T2	Course not yet completed
T3	course progress is not " terminated"
T4	on activation of DLPG
T5	Plan status ==Could be rescheduled
T6	plan status = NOT POSSIBLE
T7	Plan status is ON TRACK
T8	store course progress = terminated
T9	store course progress as completed
T10	store the updated plan in Learner records and proceed with course activity

Table 6.3: The Transitions in the generated Petri net

6.3.4 Results & Discussion

The Petri net generated for the DLPG is analyzed with the PIPE tool. The matrix analysis, Reachability graph, Steady space analysis, and State-space analysis are performed on this Petri net.

6.3.4.1 Matrix analysis

The Incidence matrix is an alternative way of describing a graph from conventional methods. 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 of the generated petrinet 6.6 are presented below.

$$
\begin{array}{c c}
 & \begin{array}{ccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
\end{array}
$$

Matrix 6.2: Forward Incidence matrix D^+ of the Petrinet

The backward incidence matrix is as follows

$$
\begin{array}{c c}
 & \begin{array}{ccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
\end{array}
$$

Matrix 6.3: Backward Incidence Matrix D^+ of the Petrinet

The combined incidence matrix D is defined as $D = D^+ - D^-$.

The combined Incidence matrix is as follows

$$
\begin{array}{cc}
 & \begin{array}{ccccccccccc} T0 & T1 & T2 & T3 & T4 & T5 & T6 & T7 & T8 & T9 & T10 \end{array} \\
\begin{array}{c} P0 \\ P1 \\ P2 \\ P3 \\ P4 \\ P5 \\ P6 \\ P7 \end{array} &
\begin{pmatrix}
0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1
\end{pmatrix}
\end{array}
$$

Matrix 6.4: Combined Incidence matrix of the petrinet

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

6.3.4.2 GSPN Analysis

The GSPN analysis presents the list of Markings or States(in Petri net terminology). There are eight markings observed for the generated Petri net. Each marking is a node in Reachability graph. The GSPN matrix is presented in Matrix 6.5.

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

Matrix 6.5: GSPN analysis matrix with list of Markings

6.3.4.3 Reachability graph for the Scenario

There are eight nodes in the Reachability graph. As the nodes are in finite number, the petrinet is bounded. The reachability graph is presented in Figure 6.7.

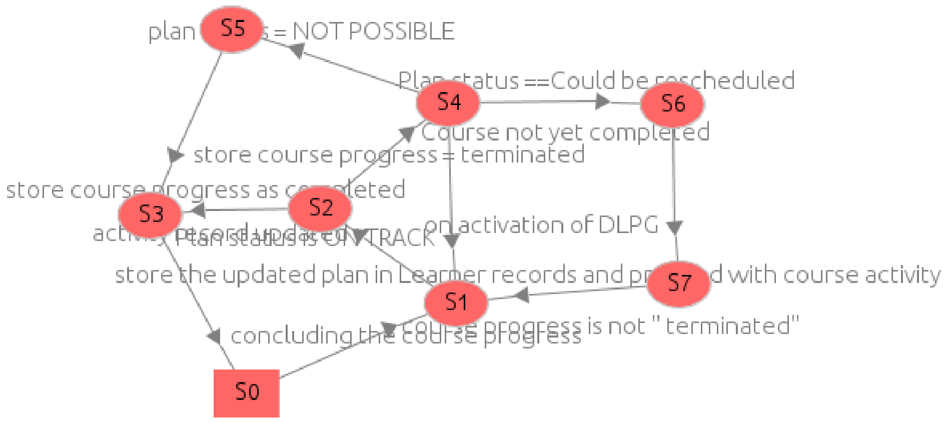


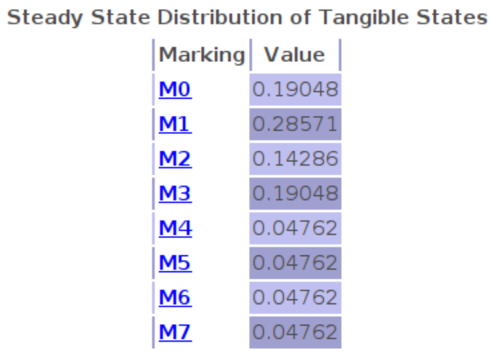
FIGURE 6.7: Reachability graph for the petrinet6.6

Each state is a node that represents a marking in a Reachability graph. There are eight states in the generated Petri net. S0 is the initial state that has the marking (0,0,0,1,0,0,0,0), which represents the place "Course Progress Status update" which initiates this scenario. If the status is not terminated, it proceeds to the next state, S1, that deals with the Record Activity(that includes verifying the activity record to check course completion). The state S2 is about the Course completion checking that leads to two more states S3, S4. The S3 is a state that confirms the course completion, and that leads to S0(once S3 →S0 is moved, there would be no more execution of this scenario as the course is terminated). The S4 is a state that analyses the plan status and takes actions as calling for a new plan (i.e., S6) or moving to a not possible state S5, which is again called S3 from there to S0. Once the plan is re-scheduled, the state rolls back to S1. A Reachability graph's significance is to identify the states with which the entire workflow can be minimally portrayed.

6.3.4.4 Steady-State Analysis

The average Steady-State distribution explains the time spent at each marking in traversing a Petri net. It is represented at 6.8.

The markings M0, M1, M2, are the repeatedly traversed nodes in the reachability graph. The Reachability graph presents that three states S0, S1, S2 are in iteration. For every iteration, the course termination and course completion must be checked.



Steady State Distribution of Tangible States

Marking	Value
M0	0.19048
M1	0.28571
M2	0.14286
M3	0.19048
M4	0.04762
M5	0.04762
M6	0.04762
M7	0.04762

FIGURE 6.8: Steady State distribution of the markings in the Petrinet

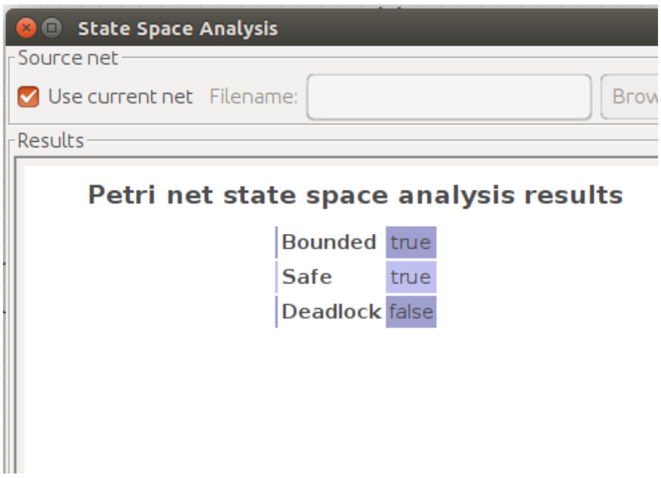


FIGURE 6.9: State space analysis for the Personalized Dynamic Learning plan generator

6.3.4.5 State Space analysis

The State space analysis determines the qualitative properties of Petri net. They are also called behavioural properties. The major conluding results are the State space analysis. The state space analysis for the DLPG scenario is presented at 6.9.

The behavioral properties confirm that the generated petrinet is safe, live and deadlock free. Thus the generated architecture is verified with this scenario.

6.4 Conclusion

The Reference Architecture with Intelligent component is validated successfully using an Architecture Description Language using OSATE. A Scenario with personalized dynamic learning plan generation is designed. Scenario-based verification is performed with formal

Petri net notation. The proposed Reference Architecture is helpful in designing a Smart Learning system with an intelligent component filter.

Chapter 7

Conclusion and Future Work

In this chapter, we summarize the results and conclusion of this thesis and also discuss the future research directions.

7.1 Conclusion

Reference Architectures are helpful for new standards development and work as a benchmark for designing the new systems. Reference Architectures for Smart Learning Environments helps in building new Smart Learning Management Systems. They contribute to the Learning Technology Systems Architecture Standard development bodies. The prime goal of this thesis is to extend the withdrawn IEEE standard for LTSA towards Smart Learning Environments. The pre-smart level IEEE LTSA is redesigned to address the Smartness standardization challenges. Three challenges are addressed in this thesis using different architecture patterns. They are summarized as follows:

First Contribution: In the first Reference Architecture, the standardization challenge "Security" is addressed. The drawbacks of centralized system is addressed in the proposed architecture. The ISO/IEC ITLET SC36 has recommended to incorporate sophisticated technologies like Blockchain into learning systems. Hence a Blockchain-enabled Reference Architecture for IEEE LTSA is proposed. It is expressed using Architecture Analysis and Description Language, which is validated using the AADL tool named OSATE. The results are validated, which facilitates the Learning System designers to proceed with pre-analysis before implementation.

The most vulnerable learning activity, "Examination" workflow is designed, and Scenario-based verification is performed. The results of the formal verification shows that the designed scenario is safe, bounded, and Deadlock-free.

Second Contribution: The Second Reference Architecture addresses the smartness standardization challenge "Interoperability." One of the limitations of the learning system to support a single Interoperable standard is overcome by adding an additional layer named "Interoperable Engine" to IEEE LTSA. This additional layer incorporates multiple interoperability standards. The layered architectural pattern is integrated into the IEEE LTSA. The proposed Reference Architecture is expressed using AADL and validated with the OSATE tool. It is observed that the architecture is successfully validated. .

An activity-based interoperability module selection scenario is designed for Scenario-based verification. The results of the formal verification with colored Petri Nets shows that the designed scenario is safe, bounded, and Deadlock-free.

Third Contribution: In the third Reference Architecture, one of the most significant smartness Standardization challenges called "Personalization" is addressed. The Smartness levels three to six requires machine intelligence. For this purpose, Pipe-filter architectural pattern is adopted in this work. An additional component named "Intelligent component" is added as a filter to the Coach's instructional delivery process. The proposed Reference Architecture is expressed and validated with AADL through the OSATE tool. The architecture is validated successfully.

A personalized assistant activity for dynamically generating the Learning plan is designed.

This activity is considered for the Scenario-based verification with Petri nets. The results of the formal verification shows that the designed scenario is safe, bounded, and Deadlock-free.

7.2 Future Directions

This section gives the directions for future work based on the work done in this thesis and the research gaps in the Smart Learning Environments.

- Apart from the three standardization challenges, the remaining Smartness Standardization Challenges can be addressed. Thus, IEEE LTSA can reach the highest Smartness Level
- One scenario for each Reference Architecture is considered in this work. There is a scope for building multiple scenarios addressing each learning activity.
- Each Reference Architecture can be considered for implementation with the organizational specific requirements
- The analysis and trade-offs can be conducted on Architecture Description Language tools after including specific inputs to the Reference Architectures
- There is scope for experimenting with all the architectural patterns that can be considered for designing new-age SLE Reference architectures
- The formal verification with Petri nets can also be attempted to implement with First-order logic too
- Several Reference Architectures can be combined in the future to get a holistic architecture that can contribute to a new standard.

References

[1] ISO/IEC JTC 1/SC 36. **Information technology for learning, education and training — A reference framework of e-Portfolio information**, 2015.

[2] Bobbio A. **System Modelling with Petri Nets.** In de Bustamante A.S. Colombo A.G., editor, *Systems Reliability Assessment. ISPRA Courses (On Reliability and Risk Analysis)*, pages 103–143. Springer, Dordrecht., 1990.

[3] Hannan Xiao Abrar Ullah and Trevor Barker. **A classification of threats to remote online examinations**. In *IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference*. IEEE, 2016.

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

[5] ADL. **SCORM Reference Introduction. Advanced Distributed Learning**.

[6] A. Alkussayer and W. H. Allen. **A scenario-based framework for the security evaluation of software architecture**. In *2010 3rd International Conference on Computer Science and Information Technology*, **5**, pages 687–695, 2010.

[7] Eka Mala Sari Rohman Andharini Dwi Cahyani, Ari Basuki and Yeni Kustiyahningsih. **Design an Adaptive E-learning Application Architecture based on IEEE LTSA reference model**. *TELKOMNIKA: Telecommunication Computing Electronics and Control*, **13**(1):284 289, 2015.

[8] Axel. **Major Centralized Systems are Hacked Multiple Times a Year**, 2018.

[9] A. Bakhouyi, R. Dehbi, M. Talea, and O. Hajoui. **Evolution of standardization and interoperability on E-learning systems: An overview**. In *2017 16th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 1–8, 2017.

[10] Len Bass, Paul Clements, and Rick Kazman. *Software Architecture in Practice*. Addison-Wesley Professional, 3rd edition, 2012.

[11] Kevin Beaver. **Hacking For Dummies, 4th Edition, Paperback**, 2013.

[12] A. Bryden. **Open and Global Standards for Achieving an Inclusive Information Society.**, 2005.

[13] W. Cai, Z. Wang, J. B. Ernst, Z. Hong, C. Feng, and V. C. M. Leung. **Decentralized Applications: The Blockchain-Empowered Software System**. *IEEE Access*, **6**:53019–53033, 2018.

[14] Pahl Claus and Barrett Ronan. **A Web Services Architecture for Learning Object Discovery and Assembly**. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & amp; Posters*, WWW Alt. '04, page 446–447, New York, NY, USA, 2004. Association for Computing Machinery.

[15] UnaMay O'Reilly Colin Taylor, Kalyan Veeramachaneni. **Likely to stop? Predicting Stopout in Massive Open Online Courses**. *CoRR*, **abs/1408.3382**, 2014.

[16] IMS Global Learning Consortium(2003a). *IMS Abstract Framework: White Paper Version 1.0.*, 2003.

[17] Alain Corbière and Christophe Choquet. **Designer integration in training cycles : IEEE LTSA model adaptation**. In Florence Michau Jean-Philippe Pernin, editor, *CALIE 2004, International Conference on Computer Aided Learning in Engineering Education*, page 12 pages, Grenoble, France, 2004. Conference website: http://www-clips.imag.fr/calie04/index.html.

[18] AICC corp. **CMI-5 - How the AICC is making eLearning better... again**.

[19] Joint Information Systems Council. *The E–Learning Framework.*

[20] M. A. A. Dewan, F. Lin, D. Wen, and Kinshuk. **Predicting Dropout-Prone Students in E-Learning Education System**. In *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pages 1735–1740, 2015.

[21] Philip Dodds. **SCORM Sequencing and Navigation, Version 1.3.1, Advance Distributed Learning, July 22, 2004.**

[22] N. CanCore Friesen. **Interoperability for Learning Object Metadata. In D. Hillman & E. L. West- brooks (Eds.), Metadata in Practice**, 2004.

[23] Francisco J. García-Peñalvo, Miguel Á. Conde, Marc Alier, and María J. Casany. **Opening Learning Management Systems to Personal Learning Environments**. *j-jucs*, **17**(9):1222–1240, may 2011.

[24] Robby Robson Geoff Collier. **Sun Microsystems white paper: e-Learning Application Infrastructure**.

[25] Begoña Gros. **The design of smart educational environments**. *Smart Learning Environments*, **3**(1):2196–7091, 2016.

[26] Faisal Ibrahim Hadiputra and Yani Widyani. **Architecture of E-Learning System based on LTSA**. In *2013 International Conference on Advanced Computer Science and Information Systems*, pages 15–20. IEEE, 2013.

[27] Tore Hoel and Jon Mason. **Standards for smart education towards a development framework**. *Smart Learning Environments*, **5**(3), 2018.

[28] Ya-Han Hu, Chia-Lun Lo, and Sheng-Pao Shih. **Developing early warning systems to predict students' online learning performance**. *Computers in Human Behavior*, **36**:469–478, 2014.

[29] Gwo-Jen Hwang. **Definition, framework and research issues of smart learning environments - a context-aware ubiquitous learning perspective**. *Smart Learning Environments*, **1**(1), 2014.

[30] IASLE. **Optimizing learning environments to enhance learning:Smart Learning**.

[31] Hazra Imran, Mohammad Belghis-Zadeh, Ting-Wen Chang, Dr Kinshuk, and Sabine Graf. **PLORS: a personalized learning object recommender system**. *Vietnam Journal of Computer Science*, **3**, 08 2015.

[32] Rafiu Isiaka, Elijah Omidiora, Stephen Olabiyisi, and Oludotun Okediran. **An Enhanced Learning Technology System Architecture for Web-Based Instructional Design**. *International Journal of Emerging Technologies in Learning (iJET)*, **11**(01):57–61, 2016.

[33] Rob Koper. **Conditions for effective smart learning environments**. *Smart Learning Environments*, **1**(1), 2014.

[34] P. Kumar, S. G. Samaddar, A. B. Samaddar, and Arun Kumar Misra. **Extending IEEE LTSA e-Learning framework in secured SOA environment**. In *2010 2nd International Conference on Education Technology and Computer*, **2**, pages V2–136–V2–140, 2010.

[35] Yiying Zhang Kun Liang, Yeshen He, Yilin Zhou, Wei Tan, and Xiaoxia Li. **Online Behavior Analysis-Based Student Profile for Intelligent E-Learning**. *Journal of Electrical and Computer Engineering*, **7**(2090-0147), 2017.

[36] Advanced Distributed Learning. *The Total Learning Architecture. Rationale and Design.*, 12 2016.

[37] Wosinski M. Liu D, Huang R. **Lecture Notes in Educational Technology**. In *Smart Learning in Smart Cities*, pages 1–240. Springer Singapore, 2017.

[38] I. Lomazova, L. Popova-Zeugmann, and A. Bartels. **Controlling boundedness for live Petri nets**. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 0236–0241, 2017.

[39] IEEE LTSC. **IEEE Standard for Learning Technology-Learning Technology Systems Architecture (LTSA)**. *IEEE Std* 1484.1 – 2003, pages 0_1 – 97, 2003.

[40] Joaquín López, Alejandro Santana-Alonso, and Miguel Díaz-Cacho Medina. **Formal Verification for Task Description Languages. A Petri Net Approach**. *Sensors*, **19**(22), 2019.

[41] Imran Makhdoom, Mehran Abolhasan, Haider Abbas, and Wei Ni. **Blockchain's adoption in IoT: The challenges, and a way forward**. *Journal of Network and Computer Applications*, **125**:251 – 279, 2019.

[42] Mason, J. Norris, and P. D. Lefrere. **An Expeditionary Approach to E-Knowledge**. In *Proceedings of EduCause in Australasia, Adelaide*, page 137–143, 2003.

[43] N. Medvidovic and R.N. Taylor. **A classification and comparison framework for software architecture description languages**. *IEEE Transactions on Software Engineering*, **26**(1):70–93, 2000. cited By 1335.

[44] Peter Mikulecky. **Smart environments for learning-multi-agent systems approach**. pages 304–310, 01 2013.

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

[46] Mairtin S O'Droma, Ivan Ganchev, and Fergal McDonnell. **Architectural and functional design and evaluation of e-learning VUIS based on the proposed IEEE LTSA reference model**. *The Internet and Higher Education*, **6**(3):263 – 276, 2003.

[47] IEEE Learning Technology Standards Committee Technical Advisory Group on xAPI. *xAPI: A Guide for Technical Implementers*, 2018.

[48] Daniel Onah, Jane Sinclair, and R Boyatt. **Dropout Rates of Massive Open Online Courses: Behavioural Patterns**. 07 2014.

[49] OWASP. **Risk rating methodology**.

[50] Spector Jonathan Michael Peng Hongchao, Ma Shanshan. **Personalized adaptive learning: an emerging pedagogical approach enabled by a smart learning environment**. *Smart Learning Environments*, **6**(1):2196–7091, 2019.

[51] Andrea Pinna, Roberto Tonelli, Matteo Orrú, and Michele Marchesi. **A Petri Nets Model for Blockchain Analysis**. *The Computer Journal*, **61**(9):1374–1388, 2018.

[52] PIPE. **PIPE: Platform Independent Petri net Editor Version 4.3.0**.

[53] Yasmina Rahmoune, Allaoua Chaoui, and Elhillali Kerkouche. **A Framework for Modeling and Analysis UML Activity Diagram using Graph Transformation**. *Procedia Computer Science*, **56**:612 – 617, 2015.

[54] Barnali Roy Choudhury and Rupinder Singh. **Learning Tools Interoperability: Paradigm shift of Open Learning**. 11 2019.

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

[56] S. Sengupta, N. Chaki, and R. Dasgupta. **Learners ' Quanta based Design of a Learning Management System**, 2009.

[57] Mike Joy Shaibu Adekunle Shonola. **Security issues in E−learning and M−learning Systems, A Comparative Analysis**. In *A Proceeding of 2nd WMG Doctoral Research and Innovation Conference (WMGRIC2015)*, 2015.

[58] Feng Haijie Shi Leyi, Li Yang. **Performance Analysis of Honeypot with Petri Nets**. *Information*, **9**(10), 2018.

[59] Elena Shoikova, Roumen Nikolov, and Eugenia Kovatcheva. **Conceptualising of Smart Education**. *SCIENTIFIC JOURNAL "ELECTROTECHNICA & ELECTRONICA"*, **3-4**:29–37, 09 2017.

[60] J. M. Spector. **Conceptualizing the emerging field of smart learning environments**. *Smart Learning Environments*, **1**(2), 2014.

[61] Yingchun Tian, Feng Chen, Hongji Yang, and Leigh Landy. **An Ontology-based Model Driven Approach for a Music Learning System**. In *Proceedings of the 21st International Conference on Software Engineering & Knowledge Engineering (SEKE'2009), Boston, Massachusetts, USA, July 1-3, 2009*, pages 739–744. Knowledge Systems Institute Graduate School, 2009.

[62] Bakken JP Uskov VL and Pandey A. **Smart Education and Smart e-Learning**. *The ontology of next generation smart classrooms*, **41**, 2015.

[63] Hambach Sybille Voskamp Jorg. **Extending the IEEE-LTSA**. In *2001 World Conference on Educational Multimedia, Hypermedia & Telecommunications. Proceedings*, pages 1967–1972. Association for the Advancement of Computing in Education (AACE), 2001.

[64] Wenxin Wu and SAEKI Motoshi. **Specifying Software Architectures Based on Coloured Petri Nets**. *IEICE Transactions on Information and Systems*, **E83D**:701–712, 06 2000.

[65] Z. Xu and W. Jian. **Research on integrated learning management system of aviation maintenance CBT based on AICC**. In *2011 International Conference on E-Business and E-Government (ICEE)*, pages 1–4, 2011.

[66] T. Okamoto S.S. Tseng Yang, S.J.H.*. **Guest Editorial: Context-aware ubiquitous learning. Educational Technology and Society**. Guest Editorial: Context-aware ubiquitous learning. Educational Technology and Society. , 11(2). 1-2. (SSCI, IF: 0.904), 2008.

[67] George Yee, Larry Korba, Nigel H. Lin, and Timothy K. Shih. **Context-aware privacy and security agents for distance education**. *International Journal of High Performance Computing and Networking*, **3**(5-6):395–404, 2005.

[68] Riezebos P. Zhu Z, Sun Y. **Introducing the smart education framework: Core elements for successful learning in a digital world.** *International Journal of Smart Technology and Learning*, **1(1)**:29–37, 2016.

Reference Architectures for Smart Learning Environments

by Anil G R

Reference Architectures for Smart Learning Environments

ORIGINALITY REPORT

21%	19%	18%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES overall Similarity : 21- (4+4+3) : 10%

1	www.ijrte.org Internet Source This publication belongs to my student	4%
2	dokumen.pub Internet Source This publication belongs to my student	4%
3	link.springer.com Internet Source This publication belongs to my student	3%
4	slejournal.springeropen.com Internet Source	2%
5	Submitted to Institute of Technology, Nirma University Student Paper	1%
6	ltsc.ieee.org Internet Source	<1%
7	grouper.ieee.org Internet Source	<1%
8	"Advances in Decision Sciences, Image Processing, Security and Computer Vision", Springer Science and Business Media LLC, 2020 Publication	<1%

9	www.informatik.uni-bremen.de Internet Source	<1 %
10	The Kluwer International Series on Discrete Event Dynamic Systems, 1998. Publication	<1 %
11	Submitted to University of Hyderabad, Hyderabad Student Paper	<1 %
12	www.repositorio.uancv.edu.pe Internet Source	<1 %
13	www.smpf.lt Internet Source	<1 %
14	www.cse.iitk.ac.in Internet Source	<1 %
15	frameworks.jtc1sc36.org Internet Source	<1 %
16	lrss.fri.uni-lj.si Internet Source	<1 %
17	"Applications and Theory of Petri Nets 2005", Springer Science and Business Media LLC, 2005 Publication	<1 %
18	Abdellah Bakhouyi, Rachid Dehbi, Mohamed Talea. "Toward an Adaptive Architecture for Integrating Mobile Apps with LMS using Next	<1 %

Generation of SCORM", 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019
Publication

19	www.forums.lvcnr.net Internet Source	<1%
20	www.iibf.deu.edu.tr Internet Source	<1%
21	ets.berkeley.edu Internet Source	<1%
22	Kefu Liu, Marek R. Kujath, Wanping Zheng. "Quantification of non-proportionality of damping in discrete vibratory systems", Computers & Structures, 2000 Publication	<1%
23	dx.doi.org Internet Source	<1%
24	eprints.soton.ac.uk Internet Source	<1%
25	Shen, Victor R.L., Cheng-Ying Yang, and Jia-Hong Lee. "Analysis of the problem description in the Algorithm for Inventive Problem Solving using Petri nets", Knowledge-Based Systems, 2012. Publication	<1%

26	deposit.ddb.de Internet Source	<1 %
27	Chandrashekar Ramanathan, Neha Gayal, Mehul Sheth, Aditya Pendyala. "A System for Managing SCORM-Compliant Learning Objects", 2011 IEEE International Conference on Technology for Education, 2011 Publication	<1 %
28	Submitted to Higher Education Commission Pakistan Student Paper	<1 %
29	Tore Hoel, Jon Mason. "Standards for smart education – towards a development framework", Smart Learning Environments, 2018 Publication	<1 %
30	www.gadaleta.org Internet Source	<1 %
31	Submitted to University of Greenwich Student Paper	<1 %
32	cent.uji.es Internet Source	<1 %
33	Submitted to Graz University of Technology Student Paper	<1 %
34	docplayer.net Internet Source	<1 %

35	edtechbooks.org Internet Source	<1 %
36	A. Paris, R. Simos, P. Andreas, S. Manolis. "Developing an architecture for the software subsystem of a learning technology system- an engineering approach", Proceedings IEEE International Conference on Advanced Learning Technologies, 2001 Publication	<1 %
37	Lecture Notes in Computer Science, 2007. Publication	<1 %
38	Paraskevi Tzouveli. "User Adaptation in E-Learning Environments", IFIP International Federation for Information Processing, 2005 Publication	<1 %
39	Lecture Notes in Computer Science, 2006. Publication	<1 %
40	Piotr Miczulski, Marian Adamski. "ANALYSIS OF SAFENESS, LIVENESS AND PERSISTENCE PROPERTIES OF PETRI NETS BY MEANS OF MONOTONE LOGIC FUNCTIONS", IFAC Proceedings Volumes, 2006 Publication	<1 %
41	researchbank.rmit.edu.au Internet Source	<1 %

42	MAJID JAMIL. "Current Status of Petri Nets Theory in Power Systems", Electric Power Components and Systems, 2004 Publication	<1 %
43	aggdefsol.com Internet Source	<1 %
44	epub.uni-regensburg.de Internet Source	<1 %
45	www.scribd.com Internet Source	<1 %

Exclude quotes	On	Exclude matches	< 14 words
Exclude bibliography	On		