An Interoperable Multi-Agent Architecture for Personalized Smart Learning Using Generative AI and Learning Analytics

Al Mahdi Khaddar, Youssef Said, Amine Dehbi, Tarik Chafiq
Laboratory of Information Processing (LTI)-Faculty of Sciences Ben M'Sick,
University Hassan II of Casablanca, Casablanca, Morocco

***Abstract*—Learning Management Systems (LMSs) remain central to digital education, but they still provide limited support for adaptive and personalized learning across heterogeneous platforms. This study proposes an interoperable smart learning architecture that integrates a Multi-Agent System (MAS), generative artificial intelligence, and learning analytics to support context-aware interventions while preserving LMS independence. Methodologically, the work follows a design-oriented research approach based on architectural modeling and scenario-based validation. The proposed framework combines Learning Tools Interoperability (LTI) 1.3, the Experience API (xAPI), Sharable Content Object Reference Model (SCORM), a Learning Record Store (LRS), and an asynchronous Extensible Messaging and Presence Protocol (XMPP)/JavaScript Object Notation (JSON) communication bus to connect intelligent services with existing LMS environments. The architecture includes tutor, assessment, recommendation, monitoring, collaboration, and profile agents coordinated through a microservices-based design. Its functional coherence is illustrated through four representative scenarios covering dropout-risk detection, targeted remediation, teacher dashboards with grade return, and collaborative feedback. The main contribution is a modular and standards-based architecture that connects analytics, agent-based orchestration, and generative AI within a closed-loop adaptation process for scalable smart learning environments.**

***Keywords*—Smart learning; multi-agent systems; generative AI; learning analytics; LMS interoperability; adaptive learning**

I. INTRODUCTION

Learning Management Systems (LMSs) such as Moodle and Canvas play a central role in digital education by supporting course delivery, communication, and assessment. However, despite their widespread adoption, most LMS platforms still provide limited support for adaptive and personalized learning at scale [1], [2]. In many cases, personalization remains restricted to static configurations, rule-based options, or isolated plugins, while intelligent services are not deeply integrated into pedagogical workflows. As higher education increasingly seeks learner-centered and data-driven environments, this limitation becomes more visible and pedagogically significant [3], [4].

At the same time, learning analytics has created important opportunities for capturing and interpreting learner behavior through digital traces. Standards such as the Experience API (xAPI) and Learning Record Stores (LRSs) enable the collection, storage, and reuse of fine-grained learning events across multiple tools and platforms [5], [6]. These technologies support monitoring, prediction, and feedback processes that can improve educational decision-making [7]. Nevertheless, in many existing systems, analytics outputs remain mainly descriptive and are not systematically transformed into timely pedagogical interventions inside the LMS [8], [9]. As a result, the adaptive loop between observation, interpretation, and educational action often remains incomplete [8].

At the same time, the use of generative artificial intelligence in education to deliver adaptive explanations, automated feedback, and personalized tutoring is gaining more attention. Several recent papers have discussed the benefits that can be offered by large language models and other artificial intelligence algorithms in improving learning processes and student engagement [10], [11], [12]. Nonetheless, many existing systems continue considering generative AI as merely another component of a conversation without embedding it into the overall educational framework along with the learning experience data, pedagogical knowledge, and limitations [13], [14]. At the same time, several challenges exist in terms of hallucination, inconsistency, trust, and pedagogical control [15], [12].

Another promising path toward intelligent learning systems would be Multi-Agent Systems (MAS). These have modularity, autonomy, and decentralized control. Agent-based technologies are applicable to many tasks including tutoring, surveillance, recommendation, evaluation, and collaborative learning [16], [17], [3]. In more recent years, microservices architecture has emerged as another promising approach to loosely coupled, scalable, and independently evolving intelligent systems [18], [19]. Many of these proposed models for education, however, are largely conceptual, platform-dependent, or disconnected from existing interoperability standards. Therefore, there remains a gap in architectural models that can integrate intelligent agents, learning analytics, artificial generative intelligence, and interoperable learning management systems [20], [13].

This fragmented landscape reveals a clear research gap. Existing works often address LMS integration, learning analytics, intelligent tutoring, agent-based adaptation, or generative AI separately, but fewer studies propose a standards-based architecture that connects these components in a closed-loop process for adaptive learning [21], [22], [7], [13]. Moreover, the literature still provides limited guidance on how such an architecture can remain modular, interoperable, and portable across heterogeneous LMS environments while supporting both learner-facing and teacher-facing interventions. This gap is particularly important in smart learning research, where the challenge is no longer only to collect educational data or deploy isolated AI services, but to orchestrate them coherently within real institutional ecosystems [23], [24].

To address this gap, this paper proposes an interoperable smart learning architecture based on a hybrid Multi-Agent System and a service-oriented microservices design. The architecture integrates Learning Tools Interoperability (LTI) 1.3 for LMS access and service delivery, xAPI and Learning Record Stores for trace collection and analytics, and asynchronous agent coordination through Extensible Messaging and Presence Protocol (XMPP) and JavaScript Object Notation (JSON). The proposed framework aims to support adaptive explanations, remediation, recommendation, monitoring, teacher dashboards, and collaborative feedback without requiring modifications to the LMS core. In this way, the LMS is preserved as a stable host environment, while intelligent services are externalized through interoperable mechanisms [6], [13].

The objectives of this study are fourfold. First, it aims to define a modular architecture that decouples intelligent services from the LMS while preserving interoperability with existing educational platforms. Second, it seeks to integrate learning analytics, intelligent agents, and generative AI into a unified closed-loop adaptation process. Third, it aims to show how standardized traces can support both individual and collaborative pedagogical interventions. Fourth, it intends to validate the internal coherence and functional completeness of the proposed architecture through representative educational scenarios.

This work adopts a design-oriented research methodology because the main goal is not to measure learner performance through a field experiment at this stage, but to design, formalize, and theoretically validate an architectural framework for smart learning. Such an approach is appropriate for complex adaptive educational systems in which interoperability, modularity, orchestration, and deployment logic must first be specified before large-scale implementation and empirical testing [25], [26], [27], [17]. Scenario-based validation is therefore used to examine whether the proposed architecture can support complete end-to-end workflows across representative educational situations, including learner support, assessment remediation, instructor analytics, and collaborative learning.

The expected contributions of this study are both scientific and practical. From a scientific perspective, the paper contributes a unified architectural model that bridges research streams that are often treated separately, namely LMS interoperability, learning analytics, intelligent agents, microservices, and generative AI in education. From a practical perspective, the proposed framework offers institutions a reusable and extensible foundation for deploying adaptive services across heterogeneous LMS ecosystems while maintaining portability, governance, and pedagogical consistency. By organizing these components within a closed-loop architecture, the study also contributes to the design of more scalable and institutionally deployable smart learning environments [23], [24].

The remainder of the paper is organized as follows. Section II reviews the related literature and identifies the main limitations of existing work. Section III presents the research methodology. Section IV describes the proposed architecture and its main layers. Section V reports the scenario-based validation and discusses the main contributions and limitations of the framework. Finally, Section VI concludes the paper and outlines directions for future work.

II. LITERATURE REVIEW

Recent research has confirmed the growing importance of adaptive and personalized learning in higher education. Personalized learning environments aim to adjust content, pacing, and support according to learner characteristics, performance, and engagement. In this context, LMS platforms remain essential for course management and digital delivery, but several studies report that most existing LMSs still provide limited support for deep and dynamic personalization [4], [21]. In many cases, adaptive behavior is not embedded in the LMS architecture itself, but added through external tools or isolated modules, which limits coherence and scalability [28], [22].

Learning analytics has become one of the most promising fields of research related to the use of digital traces for better comprehension and optimization of the learning process. Various studies demonstrate the potential of learner activity data for monitoring purposes and predictive analysis [5], [7]. At the same time, interoperability standards like xAPI and cmi5 have made collecting standard learning traces across diverse learning environments and storing them in Learning Record Stores more feasible [29], [6]. However, there are also significant limitations identified in past research. For example, many systems operate on a purely descriptive level, offering dashboard tools rather than advanced decision support, while the link between the results obtained and immediate pedagogical actions is poor [30], [8].

At the same time, artificial intelligence has gained increasing attention in smart learning systems. Recent reviews emphasize the potential of AI techniques, including machine learning, deep learning, and generative AI, to support adaptive feedback, learner modeling, performance prediction, and content personalization [10], [20]. More specifically, large language models and generative AI tools have shown promise for generating explanations, recommendations, and contextualized tutoring support [11], [12]. Nevertheless, the literature also underlines major concerns related to hallucination, factual inconsistency, bias, pedagogical reliability, and ethical use in educational settings [12], [14], [15]. Another limitation is that many AI-based educational tools are developed as standalone assistants rather than being embedded into an interoperable and institutionally deployable smart learning architecture [13].

Multi-Agent Systems constitute another important research direction for intelligent learning environments. MAS-based approaches are valued for their modularity, autonomy, and distributed problem-solving capacity, making them suitable for complex educational functions such as tutoring, assessment, recommendation, and monitoring [3], [16]. In parallel, microservice-based architectures have been recognized as effective for building scalable, loosely coupled, and evolvable intelligent systems [18], [19]. These properties are particularly relevant for educational ecosystems that need to integrate multiple intelligent services and continuously evolving AI models. However, several previous works remain either conceptual, prototype-level, or narrowly focused on one pedagogical function. In addition, many of them do not explicitly support standards-based interoperability with mainstream LMS platforms, which reduces portability and institutional applicability.

Another limitation in the existing literature is the fragmented treatment of the main components of smart learning. Some studies focus on LMS-centered personalization, others on learning analytics pipelines, others on generative AI assistants, and others on agent-based educational systems [31], [7], [13], [21]. While each stream offers useful contributions, fewer works integrate these components into a unified architecture that supports trace collection, feature computation, learner modeling, agent coordination, and delivery of interventions back into the LMS through interoperable standards. This gap becomes even more significant when both learner-facing and teacher-facing interventions are considered within the same framework.

To clarify the position of the present study, Table I compares representative previous works with the proposed approach according to key dimensions: architectural scope, interoperability support, analytics integration, agent-based intelligence, generative AI integration, and validation approach. The comparison shows that previous studies generally address only part of the problem. Some works emphasize analytics but do not include agent orchestration. Others integrate AI assistants into LMSs but do not provide a broader closed-loop adaptation model. Similarly, some studies propose intelligent educational services, yet without strong interoperability support or without considering portability across heterogeneous LMS environments [11], [13], [29], [31].

Based on this analysis, the novelty of the present work lies not in introducing a single isolated technology, but in integrating several complementary dimensions into one coherent architecture. The proposed framework combines standards-based LMS interoperability, xAPI-based trace collection, Learning Record Store-centered analytics, a hybrid Multi-Agent System implemented as microservices, and generative AI-enabled pedagogical services within a closed-loop adaptation process. In addition, the architecture is designed to support multiple educational situations, including dropout-risk detection, targeted remediation, teacher dashboards, and collaborative feedback. This combination distinguishes the proposed work from prior approaches that are typically narrower in scope, less interoperable, or insufficiently connected across analytics, intelligence, and pedagogical action [23], [24].

TABLE I. COMPARISON OF RELATED WORKS AND THE PROPOSED APPROACH

Ref.	Main Focus	LMS Interop.	Learn. Analytics	Multi Agent Supp.	Gen AI Integ.	Main Lim.
[4]	Personalized adaptive learning in HE	No explicit standards focus	Partial	No	No	Focuses on adaptive learning characteristics rather than system architecture
[7]	ML and GenAI in learning analytics	Partial	Yes	No	Yes	Reviews models and trends, but does not propose deployable architecture
[29]	Predictive analytics for course completion	Partial	Yes	No	No	Strong analytics pipeline, but no agent orchestration or LMS-wide adaptation model
[13]	GenAI assistants integrated into LMS	Yes	Partial	No	Yes	Focuses on LMS assistant integration, not full multi-agent closed-loop architecture
[11]	AI and analytics for pedagogical decisions	Partial	Yes	No	Yes	Strong decision support perspective, but limited architectural interoperability detail
[30]	Adaptive GenAI-based feedback and recommendation	No explicit standards focus	Yes	No	Yes	Focused on a specific learning system, not heterogeneous LMS environments
Proposed work	Interoperable smart learning architecture	Yes	Yes	Yes	Yes	Scenario-based validation only; empirical deployment remains future work

III. MATERIALS AND METHODS

This study adopts a design-oriented research methodology focused on the conception, formalization, and theoretical validation of an interoperable smart learning architecture. The purpose of the work is not to evaluate learner outcomes through a classroom experiment at this stage, but to develop a coherent architectural framework capable of integrating intelligent agents, learning analytics, generative artificial intelligence, and Learning Management System (LMS) interoperability within a unified smart learning ecosystem. This methodological choice is appropriate because the contribution of the paper is primarily architectural and system-oriented rather than experimental. It also responds to the need for a structured design process before empirical implementation in authentic educational environments [25].

The methodological process began with the identification of the research problem through an analysis of limitations reported in previous studies on LMS-centered personalization, learning analytics, multi-agent educational systems, and generative artificial intelligence in education. This analysis revealed a fragmented landscape in which these components are often treated separately. In particular, the literature showed the absence of a unified framework capable of combining interoperability, modularity, analytics, and intelligent pedagogical intervention within a closed-loop process. Based on this gap, the main functional and technical requirements of the proposed architecture were defined, including LMS independence, standards-based interoperability, modularity, scalability, portability across heterogeneous platforms, and support for adaptive learner-facing and teacher-facing interventions [7], [13], [21], [22], [31].

Fig. 1 presents the overall methodological workflow followed in this study, from problem identification and requirement specification to layered architecture design and scenario-based validation.

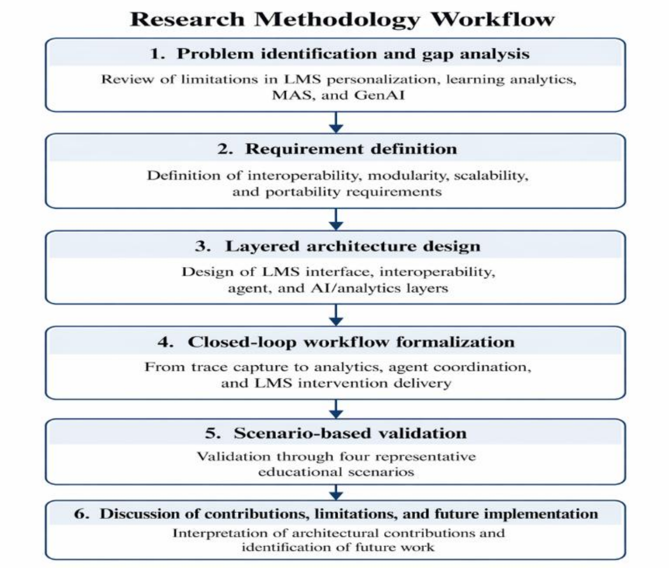

Fig. 1. Research methodology workflow.

The proposed architecture was then designed using a layered model composed of four main layers: the LMS interface layer, the interoperability layer, the intelligent agent layer, and the artificial intelligence and learning analytics layer. This decomposition was adopted to reduce coupling between components and to facilitate extensibility, maintainability, and cross-platform portability. In the proposed design, the LMS acts as a host environment that provides access to courses, users, and activities, while intelligent services are externalized and connected through interoperability standards. This architectural choice makes it possible to preserve the stability of the LMS core while enabling the integration of adaptive services developed independently of the platform. This point is particularly important because the practical value of the proposed work depends on its ability to operate across heterogeneous LMS environments without requiring changes to internal LMS code [18], [19], [27].

To specify the internal operation of the framework, the architecture was formalized as an end-to-end closed-loop workflow. Learner and teacher interactions are captured through interoperable mechanisms, represented as standardized events, and stored in a Learning Record Store. These traces are then processed by the analytics layer to compute indicators related to engagement, progression, difficulty, or risk, and to update the learner model or feature store. The resulting features are consumed by intelligent agents, which coordinate through asynchronous communication in order to produce pedagogical actions such as feedback, recommendations, remediation, alerts, or dashboard content. These actions are finally delivered back to the LMS through interoperable services. This formalization ensures that the proposed framework is not limited to a static architectural description, but includes the logical sequence of adaptive processing steps from data capture to pedagogical intervention, as requested by the reviewers [16].

At the technical level, the architecture relies on several complementary technologies and standards. Learning Tools Interoperability 1.3 is used for secure tool launch, contextual access, and LMS-facing service delivery. The Experience API is used to represent learning interactions as standardized statements, while a Learning Record Store functions as the central repository for trace collection. Intelligent services are modeled as distributed microservices, and agent coordination is supported through Extensible Messaging and Presence Protocol and JavaScript Object Notation message exchange. In addition, the analytics layer includes a feature computation process combining rule-based indicators and optional machine learning inference, depending on the educational scenario and the availability of predictive models. This technical description clarifies the implementation logic of the framework and strengthens the architectural consistency of the proposed model [6], [15], [18], [19].

The theoretical validation of the proposed architecture was conducted through a scenario-based approach. Four representative educational scenarios were defined to examine whether the framework can support complete workflows across different pedagogical situations: early dropout-risk detection, targeted remediation after quiz failure, teacher dashboard generation with grade return, and collaborative group feedback. These scenarios were selected because they cover different categories of educational intervention, namely individual monitoring, assessment support, teacher-facing analytics, and collaborative learning support. The aim of this validation is not to measure statistical performance, latency, or scalability, but to assess the functional completeness, internal coherence, and standards-based interoperability of the proposed framework across representative use cases. This clarification is important because the study is positioned as an architectural contribution and not as an empirical performance evaluation [26], [27].

Although this study does not yet include an empirical deployment in an authentic classroom setting, the adopted methodology provides a rigorous foundation for future implementation. By explicitly defining the architecture, its components, their interactions, and the logic of validation, the study establishes a reproducible design framework that can later be extended toward prototyping, simulation, and real-world experimentation. In this sense, the methodological contribution of the paper lies in transforming a fragmented set of technological and pedagogical components into a structured, interoperable, and theoretically validated smart learning architecture [25], [26], [27].

IV. PROPOSED APPROACH

This paper proposes an interoperable intelligence layer that operates independently of the Learning Management System (LMS) and enables the deployment of adaptive smart learning services across heterogeneous educational platforms. The main idea is to treat the LMS as a host environment responsible for course access, user context, and activity management, while intelligent services are externalized as interoperable components connected through standard protocols. This design allows institutions to integrate, replace, or extend intelligent educational services without modifying the LMS core, thereby improving portability, modularity, and long-term maintainability [13], [18], [19].

The proposed approach is guided by four design principles. The first principle is interoperability-first integration, which ensures that intelligent services interact with existing educational platforms through standards-based mechanisms rather than proprietary interfaces. The second principle is modularity through distributed agent microservices, which makes it possible to isolate pedagogical functions and evolve them independently. The third principle is data-driven adaptation based on standardized learning traces, allowing pedagogical interventions to be grounded in learner and group behavior. The fourth principle is governance by design, meaning that identity, access control, and privacy constraints are embedded in the architecture from the outset [6], [18].

The proposed architecture is structured into four interconnected layers: the LMS interface layer, the interoperability layer, the intelligent agent layer, and the artificial intelligence and learning analytics layer. This layered decomposition provides a clear separation of responsibilities while preserving end-to-end coordination across the system. It also enables the architecture to remain LMS-agnostic, since adaptive services are not embedded directly in LMS logic but operate through interoperable connections [13], [18], [19]. Fig. 2 illustrates an interoperable smart learning architecture based on a hybrid multi-agent system and service-oriented architecture integrating intelligent agents, generative AI, and learning analytics.

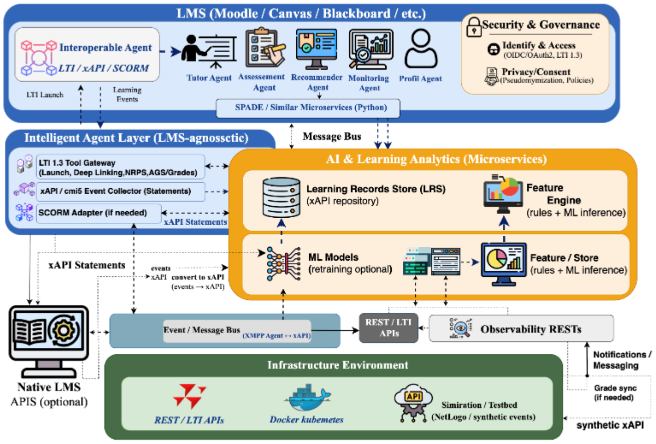

Fig. 2. Interoperable smart learning architecture based on a hybrid Multi-Agent System and Service-Oriented Architecture integrating intelligent agents, generative AI, and learning analytics.

A. Layered Architecture

The architecture is structured into four layers: 1) LMS interface, 2) interoperability, 3) intelligent agents, and 4) AI and learning analytics, deployed on a containerized infrastructure.

1) LMS interface layer: At the LMS interface layer, platforms such as Moodle, Canvas, or Blackboard remain the primary environments in which learners and instructors access courses, assessments, and collaborative activities. In the proposed model, the LMS is not redesigned at the core level. Instead, it functions as a stable delivery environment in which adaptive services are made available through external interoperable tools. This choice preserves institutional compatibility with existing LMS infrastructures while allowing intelligent functionalities to evolve outside the platform [1], [2], [13].

2) Interoperability Layer (LTI / XAPI / Cmi5 / SCORM): The interoperability layer facilitates the communication between the LMS and external intelligence services. The interoperability layer mainly uses Learning Tools Interoperability (LTI) 1.3 as the means of launching applications securely, accessing contextual data, and integrating services within the LMS environment. External tools will have access to contextual data, including the course identifier, the user's role, and activity contexts. Moreover, external tools may send back results, including grading and embedded pedagogic services, where required. Simultaneously, the Experience API (xAPI) collects learner interactions as structured statements, with CMI5 profiles available for cases where there is a need for learning activities to be packaged into structures. The learning statements are recorded in the Learning Record Store (LRS) module, enabling a consolidation of learning tracks from both the LMS and external tools. Finally, there is an optional SCORM adapter that can be managed to support legacy learning content [5], [6], [29].

3) Intelligent agent layer: At the core of the architecture, the intelligent agent layer provides the main adaptive and decision-making functions of the system. This layer is implemented as a distributed set of microservices, with each agent responsible for a specific pedagogical role. The tutor agent supports adaptive explanations, guidance, hints, and remediation messages. The assessment agent manages quiz-related diagnosis, feedback generation, and grading-oriented pedagogical logic. The recommender agent selects suitable resources, activities, or follow-up actions according to learner state and contextual needs. The monitoring agent tracks engagement, inactivity, progression, and risk indicators in order to trigger timely interventions. The collaboration agent analyzes group activity and supports collaborative learning regulation. The profile agent maintains the learner model by managing persistent features and learner descriptors used for personalization [16], [17], [18], [19].

To strengthen the technical soundness of the architecture, the decision logic of the agents is based on three complementary mechanisms. First, rule-based logic is used for explicit pedagogical conditions, such as prolonged inactivity, low quiz scores, missing milestones, or unbalanced group contribution. Second, analytics-derived features extracted from learner traces inform the interpretation of learner state, including engagement, progression, mastery, and risk indicators. Third, optional machine learning inference can be incorporated when predictive models are available, for example, to estimate dropout risk, classify engagement patterns, or rank recommended resources. In this way, agents do not operate as opaque black boxes, but as orchestrated services that consume structured inputs and produce pedagogically meaningful actions according to explicit functional roles [5], [7], [16], [29].

Agent coordination is achieved through asynchronous communication over an Extensible Messaging and Presence Protocol (XMPP) and JavaScript Object Notation (JSON) message bus. This event-driven communication model allows the agents to remain loosely coupled while supporting multi-step workflows. For example, a Monitoring Agent may publish a risk signal, which is then consumed by the Recommender Agent to select a follow-up activity and by the Tutor Agent to formulate a guidance message. This asynchronous orchestration is important in smart learning environments because it supports distributed decision-making, flexibility, and the independent evolution of services [18], [19].

4) AI and learning analytics layer: The artificial intelligence and learning analytics layer transforms raw learning traces into actionable information. The learning record store functions as the central repository of XAPI statements produced by LMS activities and interoperable tools. A feature engine processes these traces and computes pedagogically relevant indicators such as time on task, login frequency, completion rate, mastery progression, error patterns, group participation balance, and inactivity duration. These indicators are then stored in a feature store or learner model database, which provides a longitudinal representation of learners and groups. When predictive models are available, machine learning inference can be incorporated into this layer in order to generate higher-level outputs such as risk scores, recommendation rankings, or engagement classifications [29].

Generative artificial intelligence is integrated into the proposed architecture as a pedagogical support component rather than as an isolated conversational tool. Within this framework, generative AI can support adaptive explanations, remediation messages, tutoring feedback, and contextual recommendations. However, its use is constrained by pedagogical and institutional control. The output of generative services is expected to be grounded in learner context, analytics-derived features, and predefined pedagogical objectives. In addition, because generative AI may produce hallucinations or factually inconsistent outputs, the architecture assumes the use of guardrail mechanisms such as prompt restriction, context grounding, pedagogical templates, and validation rules before delivery in sensitive educational situations [10], [11], [15].

B. *End-to-End Closed-Loop Operation*

A central contribution of the proposed approach lies in the end-to-end closed-loop adaptation process. Learner and teacher interactions generate standardized traces, which are stored and analyzed to compute features and update the learner model. Intelligent agents then consume these features, coordinate with one another, and determine which pedagogical intervention should be produced. Finally, the selected intervention is delivered back into the LMS through interoperable services such as LTI tools, dashboards, recommendation panels, feedback widgets, or grade return interfaces. This closed-loop operation ensures that data collection, interpretation, decision-making, and pedagogical action are connected within the same architectural framework [6].

C. *Security, Privacy, and Institutional Constraints*

Because the architecture is designed for institutional deployment, governance and privacy constraints are considered from the outset. Identity and access control rely on secure interoperability mechanisms associated with LTI 1.3. Trace collection and learner modeling should follow data minimization principles, and pseudonymization may be applied when possible. These measures are important for ensuring that adaptive services remain compatible with institutional policies related to security, privacy, and responsible artificial intelligence use [6], [14], [26].

Overall, the proposed approach differs from narrower educational architectures by integrating LMS interoperability, standardized trace collection, distributed intelligent agents, learning analytics, and generative artificial intelligence within one coherent framework. Its novelty lies not in the isolated use of a single technology, but in the explicit orchestration of these components in a modular and closed-loop architecture capable of supporting learner-facing, teacher-facing, and collaborative interventions across heterogeneous LMS environments [24].

D. *Scenario-Based Validation*

To validate the proposed closed-loop architecture, we adopt a scenario-based validation approach. Rather than providing a statistical evaluation of performance, this approach is intended to demonstrate that the architecture is functionally complete, standards-compliant, and internally consistent across representative educational situations. Each scenario is analyzed by following an end-to-end interaction chain: 1) LTI-based access to the learning tool, 2) generation and storage of xAPI traces in the Learning Record Store (LRS), 3) feature computation and learner model update, 4) agent coordination through the message bus, and 5) delivery of adaptive interventions back to the LMS via LTI.

1) Validation method and notation: for each scenario Si, the validation process is described using a common analytical structure that captures the main functional components involved in the closed-loop workflow. As summarized in Table II, this structure includes the actors, the learning context, the observed events, the derived features, the agent decisions, and the intervention actions returned to the LMS.

TABLE II. SCENARIO VALIDATION METHOD AND NOTATION

Element	Description
Actors	Learner, instructor, LMS, and external tool or intelligent agent
Context C	Course setting, activity type, assessment constraints, and user roles
Observed events E = {e1, ⋯,en}	Learning events captured as xAPI statements

Derived features F	indicators computed by the Feature Engine through rule-based processing and, where applicable, machine learning inference
Agent decisions D	Decisions produced by the intelligent agents, along with the coordination messages exchanged through the XMPP/JSON message bus (SPADE)
Intervention actions A	Adaptive actions are delivered back to the LMS through LTI 1.3, and grades are returned through standard LTI services when required.

2) Validation Criteria: For each scenario, the proposed approach is validated through six evaluation criteria, which are detailed in Table III.

TABLE III. SCENARIO-BASED VALIDATION CRITERIA

N	criteria	Definition
C1	Traceability	Each interaction provided to the learner or tutor must be traceable through an established chain, which consists of captured events, calculated indices/attributes, and the consequent decision taken by the agent.
C2	Closed-loop completeness	It is essential that the complete adaptive cycle takes place without any gaps: context capture, event capture, learning record storage, analysis, learner model modification, decision-making by agents, and the final presentation within the LMS environment.
C3	LMS Decoupling	The scenario should not involve changes to the source code or proprietary API of the LMS. It must use interoperability standards and third-party service providers while using LMS as the hosting platform.
C4	Governance consistency	Authentication, authorization, and data access should adhere to the chosen governance strategies (authentication flows for LTI 1.3, access control) and privacy requirements (data minimization, pseudonymization if possible)
C5	Agent substitutability.	A replacement for any given agent is one that adheres to the same set of contracts: the same schema for input features, the same schema for actions, and the same message structure on the bus. Everything else in the system stays the same.
C6	Cross-platform portability	This scenario will be applicable even when there is an LMS change, as long as the new LMS has support for the same interoperability endpoints. There will be no need to alter anything apart from the configuration and registration parameters on the LMS side.

a) Scenario 1: Early dropout detection and personalized reactivation in the LMS: This scenario aims to validate that the proposed architecture can detect dropout risk at an early stage from learner activity traces and then trigger an appropriate intervention directly within the LMS. The objective is not merely to observe indicators, but to demonstrate the operation of the closed-loop adaptation cycle: interaction capture, analysis, agent-based decision-making, and delivery of a pedagogical action through an interoperable tool, as illustrated in Fig. 3.

A learner is enrolled in a course on Moodle. After an initial period of normal activity, the learner's engagement decreases (fewer logins, lack of progress, and limited interaction with learning resources). The system should identify this situation as a risk signal and propose a re-engagement action.

Step 1: Context and interaction capture (Moodle + LTI 1.3 + xAPI): The learner interacts with the course in Moodle. Intelligent services are accessed through LTI 1.3 tools, which provide secure access and learning context (course, role, activity). Relevant interactions generate learning events represented as xAPI statements (typical examples include opening an activity, viewing a resource, quiz attempt/abandonment, time spent, and progression events).

Step 2: Storage and processing (LRS + Feature Engine): The xAPI statements are stored in the Learning Record Store (LRS), which serves as the unified source of learner traces. The Feature Engine processes these traces to compute engagement indicators such as session frequency (number of logins per week), time-on-task, activity completion rate, inactivity duration (days since last interaction), and progression across course modules.

These indicators then persisted in the Feature Store / Learner Model to maintain a longitudinal state of the learner.

Step 3: Decision and orchestration (Monitoring Agent + XMPP/JSON bus): The Monitoring Agent periodically retrieves indicators from the Feature Store and applies the risk detection logic (pedagogical rules and/or ML inference, depending on the chosen design).

When risk conditions are met (e.g., prolonged inactivity and low progression), the Monitoring Agent publishes a message on the XMPP/JSON message bus (SPADE). This publication may trigger multi-agent coordination:

- The Recommender Agent selects a simple next action (a priority activity or a short resource).
- The Tutor Agent prepares a guidance/remediation message (e.g., reminder + clear instruction + suggestion for gradual re-entry).

Step 4: Delivery of the intervention in Moodle (LTI tool): The intervention is displayed directly in Moodle through an LTI component integrated into the course interface (widget, recommendation panel, or notification). The re-engagement content typically includes:

- A contextualized reminder (overdue activity, module to resume).
- A concrete recommendation (e.g., a short resource or a self-assessment exercise).
- A brief tutoring message (re-entry advice, next steps).
- Optionally, a follow-up mechanism (e.g., "I resumed" confirmation, "Start now" button).

This scenario also proves that operational adaptation is possible through this architecture, where standardized traces (xAPI) are used for engagement indicators, which then trigger an agent-based decision and eventually a direct action in the LMS through the use of LTI. The stability of the LMS is also ensured, where no internal changes are needed, and the intelligence is provided through the analytics layer and agent layer, proving the principles of decoupling and portability through other standards-based LMS systems.

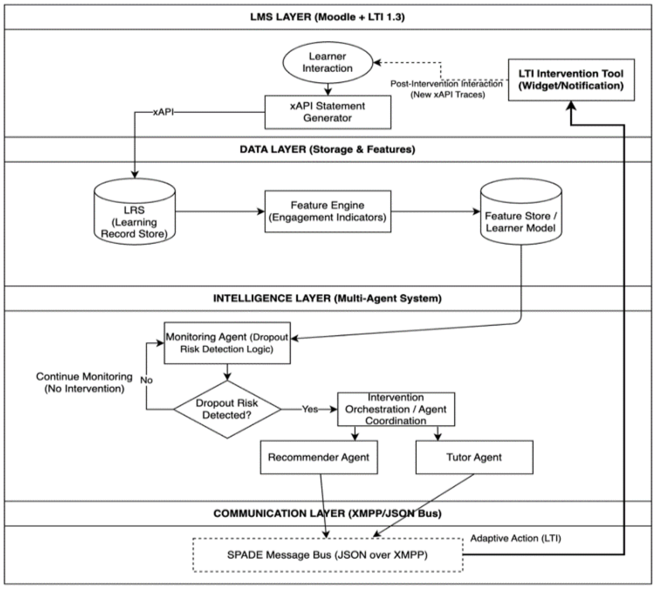

Fig. 3. Scenario 1: Early dropout risk detection and personalized re-engagement within the LMS.

b) Scenario 2: targeted remediation after quiz failure, with recommendations and tutoring in Moodle: this scenario is meant to test the fine-grained pedagogical adaptation based on assessment data. The idea is to demonstrate that assessment data, represented as standardized traces, can be used to perform a diagnostic process (e.g., poor concepts learned) and subsequently execute a specific action: providing feedback, learning materials, and tutorial support, all via lti within moodle itself, as illustrated in Fig. 4. This scenario, therefore, serves as another test of the consistency of the closed-loop process and the effectiveness of the multi-agent collaboration.

A learner is taking a quiz in Moodle or an external tool. The learner is getting a low grade or consistently answering a question incorrectly. The system must interpret the actions of the learner, determine the source of the error, and decide the best remediation.

Step 1: Capture of results and interactions (Moodle + LTI + xAPI): The learner launches the quiz, answers the questions, and either completes or abandons the attempt. Assessment-related events are captured as xAPI statements, for example: quiz start, correct/incorrect answers, final score, number of attempts, and response time.

If the quiz is associated with concepts (e.g., question tags or learning objectives), these metadata enable a more precise diagnosis.

Step 2: Storage and analysis (LRS + Feature Engine): The statements are stored in the LRS. The Feature Engine then computes indicators such as overall score, error rate by concept (when available), most problematic questions, abnormally high or low response time, and progression across attempts.

These data update the learner model/feature store, including the estimated mastery level for each concept, recurrent difficulties, and progression state.

Step 3: Decision and orchestration (Assessment Agent, Tutor Agent, Recommender Agent): An assessment agent, or the analysis logic itself, detects a remediation condition (e.g., low mastery of a concept). Then, through the XMPP/JSON bus (SPADE):

- The recommender agent selects resources adapted to the weak concept (lesson, short video, simple exercise),
- The tutor agent generates pedagogical feedback (explanation of the typical error, concept reminder, worked example),
- Optionally, the monitoring agent records that the learner is having trouble, enabling longitudinal follow-up.

Step 4: Delivery of remediation in Moodle (LTI tool): In Moodle, an LTI panel displays:

- A short explanatory feedback message (what was not understood).
- A prioritized recommendation of resources.
- A reinforcement activity (mini-exercise, short quiz, checklist).
- A suggested recovery strategy (e.g., review section X, then retry the quiz).

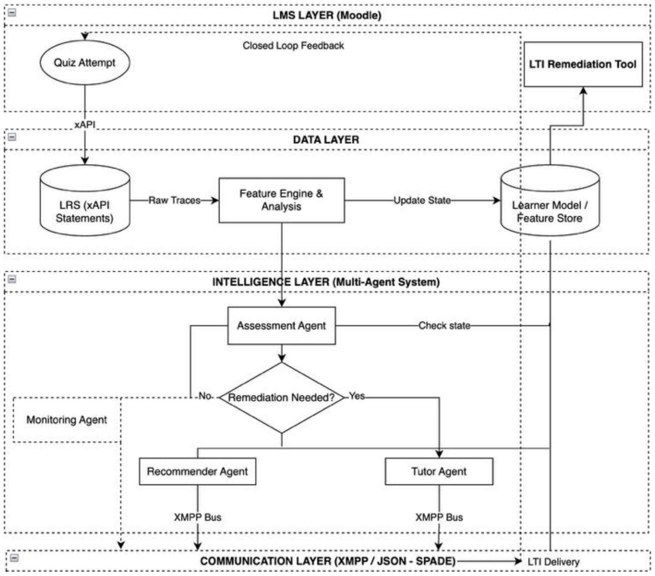

Fig. 4. Scenario 2: Targeted remediation after quiz failure, with recommendations and tutoring in Moodle.

In this scenario, architecture can take the results of assessment and translate them into active pedagogical action without the need for an LMS. This is because the same agents and analytics services could be utilized with any standards-based LMS platform due to the input/output remaining the same.

c) Scenario 3: Teacher dashboard and grade return: the above situation serves to validate the institutional and operational aspects of the proposed approach. In addition to the learner-focused services, the system needs to offer aggregated analytics that are relevant to the instructor and grade return via standard services when needed. As illustrated in Fig. 5, architecture enables the collection, analysis, and visualization of assessment results to support pedagogical decision-making at the course level.

The assessment activity may be a quiz, an assignment, or an external learning activity. The results are individual. The teacher needs a dashboard to get an overall picture of how the class is performing. Additionally, the teacher needs a dashboard to identify areas where the class may be struggling. Finally, the teacher needs a dashboard to automatically sync the results back into the Moodle site if the assessment activity is an external tool.

Step 1: Capture of results (LTI + xAPI): The assessment can be a native one or an external one via an LTI 1.3-enabled tool. The results can be scored, successful or unsuccessful, progression, time spent, or attempts. These are then sent to the central LRS.

Step 2: Computation of cohort metrics (Feature Engine + analytics): The analytics engine computes class-level indicators, such as:

- Mean, median, and score distribution,
- Success rate by question or concept,
- Difficulty indicators,
- Detection of at-risk groups (e.g., learners showing low progression),
- Correlation between engagement and performance (when available).

These metrics are stored in the Feature Store and made available to visualization services.

Step 3: Dashboard generation (dashboard service): A dashboard service can be used to obtain the aggregated metrics, which are used to create a view for the teacher. This view is embedded within the LMS as a tool using the LTI, so the teacher can work within the familiar environment. The dashboard can display:

- Overall cohort progression.
- Problematic concepts.
- A list of learners requiring follow-up.
- Pedagogical recommendations (e.g., revisiting concept X in class).

Step 4: Grade return to Moodle (LTI services): If the assessment activity is externalized, the grades can be sent back to Moodle via the usual services provided by LTI (grade synchronization when needed). The grade is then visible in the gradebook without any manual intervention.

This scenario validates the architecture's ability to support institutional needs: cohort analytics, instructor-facing interfaces integrated into the LMS, and standards-based grade management. It strengthens the argument for deployable architecture in real educational environments, where governance, LMS stability, and interoperability are essential.

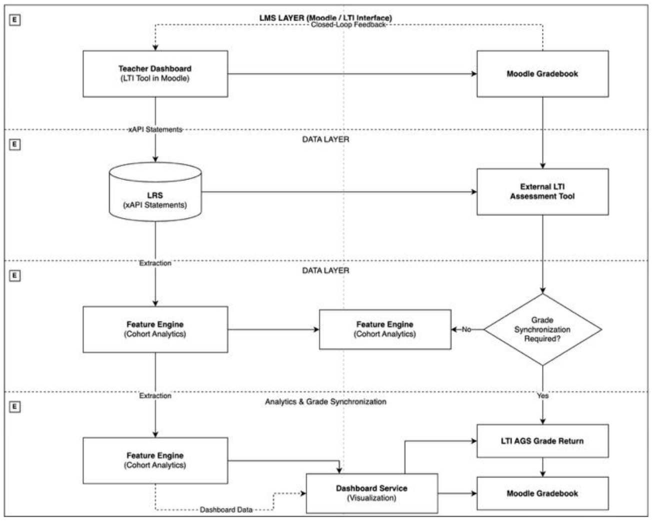

Fig. 5. Scenario 3: Teacher dashboard and grade return.

d) Scenario 4: Group interaction analysis and collaborative feedback integrated into Moodle: This scenario is intended to validate that not only is this architecture applicable for individual adaptations, but also that it is applicable for collaborative learning situations, including group work, forums, projects, etc. The purpose is to validate that traces of interaction between learners can be gathered in a standard format, analyzed to detect patterns of collaboration (such as unbalanced participation, no coordination, low levels of interaction), and then converted into pedagogical interventions that are directly provided inside Moodle through LTI tools, as illustrated in Fig. 6.

The learners are working on a collaborative activity such as a project, forum discussion, wiki, document, or integrated external tool. Some groups may be successful, while others may not be successful. This may be shown by one learner completing the entire project, low overall participation, low levels of communication among the group members, or the group failing to meet the deadlines. The system needs to recognize such groups and give them feedback.

Step 1: Capture of collaborative interactions (Moodle + tools + xAPI): Actions related to collaboration are recorded in the form of events like posts, replies, comments, file submissions, document editing, messages, validation of milestones, meetings, etc. These actions are then sent out in the form of xAPI statements. The traces may originate from:

- Native Moodle tools (forum, assignment, group activity).
- Or external tools integrated through LTI 1.3.

Step 2: Storage and analysis (LRS + Feature Engine): The LRS centralizes the statements. The Feature Engine computes individual- and group-level indicators, for example:

- Contribution distribution across group members.
- Interaction and reply frequency within the group.

- Response delays.
- Regularity of participation over time.
- Progress with respect to project milestones.
- Coordination indicators (activity concentrated on one learner, periods of collective inactivity).

These indicators feed the feature store (group profile and individual profiles), enabling longitudinal analysis.

Step 3: Decision and orchestration (Collaboration Agent + Monitoring Agent): The collaboration agent exploits group-level metrics to detect patterns such as:

- Unbalanced participation (one dominant contributor).
- Low overall activity (inactive group).
- Lack of exchanges (few replies or interactions).
- Risk of delay (slow milestone progression).

When no critical pattern is detected, the system continues monitoring the group without triggering intervention. When a relevant pattern is detected, the Collaboration Agent initiates the intervention flow through the SPADE bus (XMPP/JSON). Depending on the case, this may coordinate:

- The Monitoring Agent to generate a "group at risk" alert.
- The Tutor Agent to formulate collaborative feedback or guidance.
- Optionally, the Recommender Agent suggests resources on project management, role distribution, or coordination strategies.

Step 4: Delivery of feedback in Moodle (LTI tools): The system delivers the intervention through an LTI tool integrated into Moodle. Once the intervention has been coordinated through the SPADE bus, the Feedback Delivery Service receives the outputs produced by the agents, aggregates the relevant alerts, recommendations, and tutoring feedback, and transforms them into an LTI-compatible payload. It then packages the pedagogical action into a deliverable format and transmits it to the LTI Intervention Tool for direct presentation within Moodle. Depending on the pedagogical policy, the intervention may be:

- Addressed to the group: feedback message, coordination advice, task checklist, role distribution suggestion,
- Addressed to the teacher: group dashboard, alerts on at-risk teams, contribution summary,
- Or mixed: lightweight feedback to the group plus a detailed alert to the teacher.

This scenario also proves that architecture supports collaboration analytics/intervention without relying on a particular LMS solution. The traces are standardized through xAPI, and the delivery is done through LTI, making this mechanism easily transportable to other standards-based LMS environments. This scenario also makes the model more relevant to institutions, since the teacher can gain insight into group collaboration.

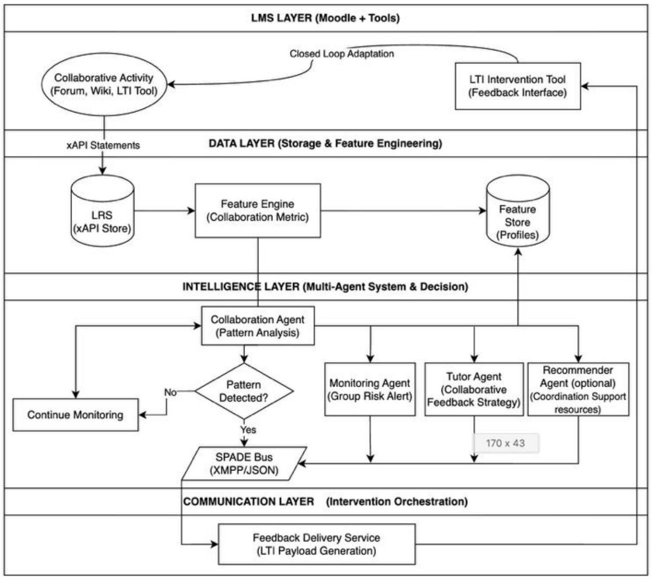

Fig. 6. Scenario 4: Group interaction analysis and collaborative feedback integrated into Moodle.

V. RESULT AND DISCUSSION

The results of this study are architectural and analytical rather than experimental. Accordingly, the discussion focuses on the main findings derived from the design and scenario-based validation of the proposed framework. The analysis shows that the proposed architecture contributes in four main ways: it improves interoperability across heterogeneous Learning Management System (LMS) environments, integrates analytics and intelligent agents within a closed-loop workflow, incorporates generative artificial intelligence under pedagogical and governance constraints, and supports multiple categories of educational intervention through a unified modular design.

A first important finding is that the proposed framework improves LMS decoupling and interoperability compared with many existing intelligent learning solutions. In several previous works, adaptive functions remain strongly tied to a specific platform, tool, or plugin, which limits portability and long-term maintainability. In contrast, the proposed architecture separates the LMS hosting role from the intelligence layer and relies on standards such as Learning Tools Interoperability (LTI) 1.3, the Experience API (xAPI), cmi5, and Sharable Content Object Reference Model (SCORM) to ensure communication between the LMS and external intelligent services. This design reduces platform dependency and makes it possible to deploy the same adaptive logic across multiple LMS environments without modifying the LMS core. From an institutional perspective, this is a significant advantage because it supports reuse, extensibility, and more sustainable integration of smart learning services.

A second finding is that the proposed model offers a stronger integration between learning analytics and pedagogical action than many related architectures. Several previous studies have emphasized data collection, dashboards, or predictive analytics, but fewer have connected these components to a complete adaptive intervention cycle inside the LMS [32]. In the proposed framework, learner and teacher interactions are captured as standardized traces, stored in a Learning Record Store, transformed into pedagogically relevant indicators, and then consumed by intelligent agents that generate contextualized interventions. This closed-loop organization constitutes an important contribution because it links observation, interpretation, decision-making, and action within one coherent architecture. In this sense, the framework goes beyond descriptive analytics and moves toward operational adaptation.

A third finding concerns the role of intelligent agents in structuring adaptive decision-making. The proposed framework defines a distributed set of agents with complementary pedagogical roles, including tutoring, assessment support, recommendation, monitoring, collaboration analysis, and learner modeling. Compared with narrower agent-based educational systems, the proposed architecture does not restrict intelligence to a single function. Instead, it supports multi-agent orchestration through asynchronous communication and shared analytics outputs. The inclusion of explicit rule-based logic, analytics-derived features, and optional machine learning inference also strengthens the technical soundness of the model. This means that the intelligent agents are not only conceptual entities, but are embedded in a structured decision process that can support explainable and context-aware interventions.

A fourth finding is that the integration of generative artificial intelligence is more pedagogically controlled than in many current educational AI solutions. In numerous existing approaches, generative artificial intelligence is introduced mainly as a standalone assistant or conversational interface. In contrast, the proposed framework positions generative services within a broader architecture governed by learner context, analytics-derived features, pedagogical objectives, and institutional rules. In addition, the model explicitly acknowledges risks related to hallucination and factual inconsistency and assumes the use of guardrail mechanisms such as prompt restriction, context grounding, pedagogical templates, and validation rules. This aspect is important because it addresses a central concern raised in the recent literature and by the reviewers, namely that generative artificial intelligence should not be integrated into educational settings without mechanisms that support reliability and pedagogical control.

The scenario-based validation also highlights the versatility of the proposed architecture. Scenario 1 shows that the framework can support early dropout-risk detection and re-engagement by linking engagement indicators to personalized interventions delivered directly within the LMS. Scenario 2 demonstrates fine-grained remediation after quiz failure through the combination of assessment traces, diagnosis, tutoring feedback, and targeted recommendations. Scenario 3 extends the framework toward teacher-facing analytics and grade return, which is important for institutional deployment because smart learning systems must serve not only learners but also instructors. Scenario 4 shows that the architecture can support collaborative learning by analyzing group interactions and delivering group-oriented or teacher-oriented feedback. Taken together, these four scenarios indicate that the same architectural principles can support individual, collaborative, learner-facing, and teacher-facing interventions within one coherent system.

When compared with representative related works, the proposed architecture shows several advantages. Unlike studies centered mainly on adaptive learning characteristics, the present work provides an explicit architectural model. Unlike analytics-focused studies, it includes agent-based orchestration and direct intervention delivery. Unlike LMS-integrated artificial intelligence assistants, it extends beyond a single assistant paradigm and supports a broader multi-agent ecosystem. Unlike function-specific recommendation or feedback systems, it is designed for heterogeneous LMS environments and incorporates standards-based interoperability as a central principle. Therefore, the novelty of this study lies not in the isolated use of one technology, but in the integration of interoperability standards, learning analytics, distributed intelligent agents, and generative artificial intelligence within a modular closed-loop smart learning architecture.

Despite these contributions, the study also has important limitations. First, the validation remains theoretical and scenario-based. Although this approach is appropriate for assessing functional completeness and architectural coherence at this stage, it does not provide empirical evidence regarding performance, latency, scalability, usability, or educational effectiveness. Second, the operational deployment of a distributed architecture combining analytics, intelligent agents, and generative artificial intelligence may introduce computational and governance challenges in real institutional settings. Third, although the framework includes conceptual safeguards for responsible artificial intelligence use, these mechanisms still need to be implemented and evaluated in authentic contexts. These limitations confirm that the present work should be understood as an architectural contribution and a foundation for future experimentation rather than as a fully validated deployment model.

Overall, the discussion confirms that the proposed framework provides a coherent, extensible, and interoperable contribution to smart learning research. Its main added value lies in the ability to connect standardized trace collection, learner modeling, intelligent agent orchestration, and pedagogically grounded interventions within a unified architecture that remains portable across heterogeneous LMS platforms. In this way, the study contributes not only to the conceptual design of next-generation smart learning environments but also to the definition of a reproducible architectural basis for future prototype development and empirical implementation.

VI. CONCLUSION

This study proposed an interoperable smart learning architecture designed to support adaptive and personalized learning across heterogeneous Learning Management System (LMS) environments. The work addressed the research gap identified in the literature by defining a modular framework that decouples intelligent services from the LMS while preserving standards-based interoperability. In this sense, the first objective of the study was achieved through the design of an architecture that separates the LMS hosting role from the intelligence layer and enables adaptive services to operate through interoperable protocols rather than through modifications of the LMS core.

The second objective of the study was to integrate learning analytics, intelligent agents, and generative artificial intelligence into a unified closed-loop adaptation process. The proposed framework achieved this by connecting trace collection, feature computation, learner modeling, agent-based orchestration, and intervention delivery within one coherent architecture. This integration represents one of the main contributions of the work because many existing approaches address these dimensions separately rather than as part of a coordinated educational ecosystem.

The third objective was to show how standardized learning traces can support both individual and collaborative pedagogical interventions. Through the four scenario-based validations, the study demonstrated that the same architectural logic can support early dropout-risk detection, targeted remediation after quiz failure, teacher-facing dashboards with grade return, and collaborative feedback for group learning situations. These scenarios highlight the functional completeness of the architecture and show its ability to support learner-facing, teacher-facing, individual, and collaborative services within the same framework.

The novelty of this work lies not in the isolated use of a single technology, but in the explicit integration of interoperability standards, learning analytics, distributed intelligent agents, and generative artificial intelligence within a modular and closed-loop smart learning architecture. Compared with narrower related works, the proposed framework places stronger emphasis on LMS independence, agent orchestration, pedagogically grounded analytics, and institutional portability across heterogeneous platforms. This gives the model a broader architectural scope and a stronger deployment perspective than many existing intelligent learning solutions.

From a scientific perspective, the study contributes a reproducible architectural foundation for smart learning research by bringing together research streams that are often treated separately. From a practical perspective, the framework offers institutions a structured basis for deploying adaptive services in existing LMS ecosystems while preserving governance, portability, and compatibility with standards-based infrastructures. In this way, the proposed model may support the future development of more scalable, flexible, and institutionally reliable smart learning environments.

At the same time, the study has important limitations. The validation remains theoretical and scenario-based, which means that the work does not yet provide empirical evidence regarding usability, performance, scalability, or educational effectiveness in authentic classroom settings. In addition, the implementation of intelligent agents and generative artificial intelligence in real institutional environments may raise technical, ethical, and governance-related challenges that require further investigation. These limitations indicate that the present study should be understood primarily as an architectural and conceptual contribution rather than a fully validated deployment model.

Future work should therefore focus on empirical implementation and evaluation in authentic educational contexts. This includes prototype development, experimentation in real LMS platforms, measurement of system performance and user experience, and assessment of the educational effectiveness of agent-based and generative artificial intelligence-supported interventions. Further research may also explore explainable and trust-aware agent behavior, more advanced multimodal learning analytics, and stronger governance mechanisms for privacy, ethics, and responsible artificial intelligence in smart learning systems.

REFERENCES

[1] Munna, M. S. H., Hossain, M. R., & Saylo, K. R. (2024). Digital education revolution: Evaluating LMS-based learning and traditional approaches. *Journal of Innovative Technology Convergence*, *6*(2). https://doi.org/10.69478/JITC2024v6n002a03

[2] Dehbi, A., Bakhouyi, A., Dehbi, R., & Talea, M. (2022). MOOCs in smart education: Comparative study by applying AHP and COPRAS method. *International Journal of Emerging Technologies in Learning (iJET)*, *17*(8), 61-74. https://doi.org/10.3991/ijet.v17i08.27871

[3] Abu-Rasheed, H., Weber, C., & Fathi, M. (2023, July). Context based learning: a survey of contextual indicators for personalized and adaptive learning recommendations–a pedagogical and technical perspective. In *Frontiers in Education* (Vol. 8, p. 1210968). Frontiers Media SA. https://doi.org/10.3389/feduc.2023.1210968

[4] Du Plooy, E., Casteleijn, D., & Franzsen, D. (2024). Personalized adaptive learning in higher education: A scoping review of key characteristics and impact on academic performance and engagement. *Heliyon*, *10*(21). https://doi.org/10.1016/j.heliyon.2024.e39630

[5] Alalawi, K., Athauda, R., & Chiong, R. (2025). An extended learning analytics framework integrating machine learning and pedagogical approaches for student performance prediction and intervention. *International Journal of Artificial Intelligence in Education*, *35*(3), 1239-1287. https://doi.org/10.1007/s40593-024-00429-7

[6] Miller, B., Rutherford, T., Pack, A., & Johnson, A. (2021, November). Bridging the SCORM and xAPI Gap: The Role of cmi5. In *Interservice/Industry Training, Simulation, and Education Conference, November-December* (pp. 1-11).

[7] Rodríguez-Ortiz, M. Á., Santana-Mancilla, P. C., & Anido-Rifón, L. E. (2025). Machine learning and generative AI in learning analytics for higher education: A systematic review of models, trends, and challenges. *Applied Sciences*, *15*(15), 8679. https://doi.org/10.3390/app15158679

[8] Ngulube, P., & Ncube, M. M. (2025). Leveraging Learning Analytics to Improve the User Experience of Learning Management Systems in Higher Education Institutions. *Information*, *16*(5), 419. https://doi.org/10.3390/info16050419

[9] Samuelsen, J., Chen, W., & Wasson, B. (2021). Enriching context descriptions for enhanced LA scalability: a case study. *Research and Practice in Technology Enhanced Learning*, *16*(1), 6. https://doi.org/10.1186/s41039-021-00150-2

[10] Hariyanto, Kristianingsih, F. X. D., & Maharani, R. (2025). Artificial intelligence in adaptive education: a systematic review of techniques for personalized learning. *Discover Education*, *4*(1), 458. https://doi.org/10.1007/s44217-025-00908-6

[11] Sajja, R., Sermet, Y., Cwiertny, D., & Demir, I. (2025). Integrating AI and learning analytics for data-driven pedagogical decisions and personalized interventions in education. *Technology, knowledge and learning*, 1-31. https://doi.org/10.1007/s10758-025-09897-9

[12] Qian, Y. (2025). Pedagogical Applications of Generative AI in Higher Education: A Systematic Review of the Field. *TechTrends*, 1-16. https://doi.org/10.1007/s11528-025-01100-1

[13] Alier, M., Pereira, J., Garcia-Penalvo, F. J., Casan, M. J., & Cabre, J. (2025). LAMB: An open-source software framework to create artificial intelligence assistants deployed and integrated into learning management systems. *Computer Standards & Interfaces*, *92*, 103940. https://doi.org/10.1016/j.csi.2024.103940

[14] Tu, Y., Chen, J., & Huang, C. (2025). Empowering personalized learning with generative artificial intelligence: Mechanisms, challenges and

pathways. *Frontiers of Digital Education*, *2*(2), 1-18. https://doi.org/10.1007/s44366-025-0056-9

[15] Elsayed, H. (2024). The impact of hallucinated information in large language models on student learning outcomes: A critical examination of misinformation risks in AI-assisted education. *Northern Reviews on Algorithmic Research, Theoretical Computation, and Complexity*, *9*(8), 11-23. https://doi.org/10.3390/app15158679

[16] El Akhdar, A., Baidada, C., & Kartit, A. (2024, April). Adaptability of Microservices Architecture in IoT Systems: A Comprehensive Review. In *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security* (pp. 1-9). https://doi.org/10.1145/3659677.3659734

[17] Dehbi, A., Bakhouyi, A., Dehbi, R., & Talea, M. (2024, December). Towards a Smart Learning Environment: A Model for Smart Education using IoT and AI. In *2024 6th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)* (pp. 1-5). IEEE. https://doi.org/10.1109/ISAECT64333.2024.10799743

[18] Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. *Present and ulterior software engineering*, 195-216. https://doi.org/10.1007/978-3-319-67425-4_12

[19] Moreschini, S., Pour, S., Lanese, I., Balouek, D., Bogner, J., Li, X., ... & Taibi, D. (2025). AI Techniques in the Microservices Life-Cycle: a Systematic Mapping Study. *Computing*, *107*(4), 100. https://doi.org/10.1007/s00607-025-01432-z

[20] Tan, L. Y., Hu, S., Yeo, D. J., & Cheong, K. H. (2025). Artificial Intelligence-Enabled Adaptive Learning Platforms: A Review. *Computers and Education: Artificial Intelligence*, 100429. https://doi.org/10.1016/j.caeai.2025.100429

[21] Strielkowski, W., Grebennikova, V., Lisovskiy, A., Rakhimova, G., & Vasileva, T. (2025). AI-driven adaptive learning for sustainable educational transformation. *Sustainable Development*, *33*(2), 1921-1947. https://doi.org/10.1002/sd.3221

[22] Molinari, A., & Sandri, S. (2024, November). Evolution of lms design and implementation in the age of ai and large language models. In *Proceedings of the Second International Workshop on Artificial INtelligent Systems in Education co-located with 23rd International Conference of the Italian Association for Artificial Intelligence (AIxIA 2024), Bolzano, Italy*.

[23] Bakhouyi, A., Dehbi, A., Amhaimar, L., Broumi, S., Talea, M., & Khalidi, A. (2025). Improving SMART learning: Course completion via AI-driven hybrid system integration in big data. Telematics and Informatics Reports, 18, 100199. https://doi.org/10.1016/j.teler.2025.100199

[24] Khaddar, A. M., Dehdi, A., Alali, A., Bakhouyi, A., & Talea, M. (2025). Emerging Technologies in Learning: A Bibliometric Analysis of Technology Integration and Applications. International Journal of Interactive Mobile Technologies, 19(5). https://doi.org/10.3991/ijim.v19i05.51267

[25] Gilbert, N., & Troitzsch, K. (2005). Simulation for the social scientist. McGraw-Hill Education (UK).

[26] Railsback, S. F., & Grimm, V. (2019). *Agent-based and individual-based modeling: a practical introduction*. Princeton university press.

[27] Dehbi, A., Bakhouyi, A., Dehbi, R., & Talea, M. (2024). Smart evaluation: a new approach improving the assessment management process through cloud and IoT technologies. *International journal of information and education technology*, *14*(1), 107-118. https://doi.org/10.18178/ijiet.2024.14.1.2030

[28] Yan, L., Echeverria, V., Jin, Y., Fernandez-Nieto, G., Zhao, L., Li, X., ... & Martinez-Maldonado, R. (2024). Evidence-based multimodal learning analytics for feedback and reflection in collaborative learning. *British Journal of Educational Technology*, *55*(5), 1900-1925. https://doi.org/10.1111/bjet.13498

[29] Bakhouyi, A., Dehbi, A., Amhaimar, L., Tazouti, Y., Nadir, Y., & Khalidi, A. (2025). Enhancing learning outcomes in smart education: A supervised machine learning predictive analytics model for course completion. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 14(6), 4711–4721. https://doi.org/10.11591/ijai.v14.i6.pp4711-4721

[30] Nongkhai, L. N., Wang, J., Wynn, A., & Mendori, T. (2025). Evaluating adaptive and generative AI-based feedback and recommendations in a knowledge-graph-integrated programming learning system. *Computers and Education: Artificial Intelligence*, 100526. https://doi.org/10.1016/j.caeai.2025.100526

[31] Mazzullo, E., Bulut, O., Wongvorachan, T., & Tan, B. (2023). Learning analytics in the era of large language models. *Analytics*, *2*(4), 877-898. https://doi.org/10.3390/analytics2040046

[32] Dehbi, A., Bakhouyi, A., Dehbi, R., & Talea, M. Towards Technology-Enhanced Learning: A Novel Machine Learning Approach in Education 4.0. *Available at SSRN 4974380*. https://doi.org/10.1016/j.teler.2025.100194