Design and Modeling of a Dynamic Adaptive Hypermedia System Based on Learners' Needs and Profile

Mohamed Benfarha[1]*, Mohammed Sefian Lamarti[2], Mohamed Khaldi[3]
A Research Team in Computer Science and University Pedagogical Engineering-Applied Mathematics and Computer Science (AMCS) / Higher Normal School of Tétouan, Abdelmalek Essaadi University, Morocco[1, 2]
A Research Team in prt, Abdel Malek Essaadi University, Morocco[3]

***Abstract*—This study presents the design and modeling of an adaptive hypermedia system, capable of dynamically adjusting to the needs and characteristics of each learner according to their profile. In the digital age, where digital content must respond to varied profiles and adapt to learners' preferences and skills, this system offers a personalized approach that improves the learning and interaction experience. This personalized approach aims to enrich the learning and interaction experience with learning environments. This work consists of analyzing the different types of learner profiles, in order to identify the key criteria for effective personalization. Based on this, the authors developed a model of an adaptive and dynamic hypermedia system, capable of adapting in real time. To ensure a clear and coherent structure, the use of UML (Unified Modeling Language) modeling is increased. Preliminary results show that this system offers a relevant and targeted experience thanks to learner engagement and satisfaction, making learning both more relevant and more enjoyable. This work paves the way for future research on the optimization of hypermedia systems by further integrating the individual behaviors of learners, in a truly adaptive learning environment, which values the potential of each learner.**

Keywords—Design; adaptive hypermedia; learning styles; user modeling; UML models

I. Introduction

Currently, hypermedia learning systems, whether online educational platforms, interactive digital courses or distance learning tools, interactive digital environments occupy a central place in modern education [1], [2]. By combining texts, images, videos, sounds and hypertext links to enrich the user experience [3]. These systems offer a richer learning experience than traditional media. However, despite their advantages, the majority of these systems lack sufficient adaptability to personalize the experience according to users, their context and their learning styles [4]. Most of them have a major limitation: their rigidity. Few are able to truly adapt to the particularities of each learner, their learning style, their skill level or their context of use. Faced with this observation, dynamic adaptive learning systems are emerging as a promising solution and represent a significant evolution in online learning and knowledge management [5]. The integration of artificial intelligence techniques, recommendation technologies in platforms and recommendation algorithms, allows to meet the growing need for individualization in education and takes into account the individual differences of users, their learning preferences, their skill levels and their context of use, while offering personalized, effective and optimized learning experiences and adjusting the content and interface in real time to improve the learning experience and effectiveness. Personalize the educational experience in real time, adjusting the content, the difficulty of the exercises, and even the user interface to optimize engagement and learning effectiveness [6], [7]. This individualized approach meets a growing need in the educational field, where learners have different rhythms, preferences and objectives [8], [9]. However, designing such a system requires a flexible architecture and complex decision logic, capable of processing heterogeneous data (such as user performance, interactions or history) to generate relevant learning paths [10], [11]. This is where Unified Modeling Language (UML) plays a key role. By providing a visual and structured representation of system components, interactions between actors, and adaptation mechanisms, UML allows these technical requirements to be formalized while ensuring a robust and scalable design [12], [13]. Using UML for the design of personalized and flexible educational systems is crucial. Thanks to a clear representation of the dynamic structure, complex interactions, and adaptation mechanisms, UML allows the specification of both functional requirements such as content customization and non-functional requirements such as performance and security [14], [15], [16], [11], [17], [18], [19], [20], [21] and [22].

This research aims to design an innovative model of a dynamic adaptive hypermedia system capable of adjusting in real time to the specific needs of users. To achieve this, the working approach in this study is based on a UML modeling defining the components, interactions and adaptation mechanisms necessary for the personalization of the user experience in real time and complete that integrates three essential dimensions: the management of learner profiles (skills, preferences and learning history), the adaptation mechanisms (personalization rules and content recommendation algorithms), and the underlying software architecture (functional modules, data flows and user interfaces). The study is structured in four main parts: we will start by drawing up an in-depth state of the art of existing adaptive hypermedia systems and their current limitations; we will then present our design methodology with the different UML diagrams (use cases, classes and sequences); then we will analyze the preliminary results demonstrating the impact of adaptivity on learner engagement; Finally, we will explore future perspectives, particularly the advanced integration of artificial intelligence to refine personalization. Through this study, we aspire to contribute significantly to the development of intelligent learning environments where technology serves the development of each learner's individual potential.

This research adds to recent studies on adaptive hypermedia systems [4], [6], [8], while presenting notable methodological advances. Unlike current methods that rely mainly on adjustment based on fixed profiles [9], [12] or established guidelines [15], our model proposes a more advanced dynamic approach, drawn from recent advances in the field of educational AI [17], [20]. In contrast to traditional approaches that modify content linearly [5], [7], our device incorporates a double adaptation: both of learning trajectories and user interfaces, thus filling two gaps identified in the publications [10], [13]. Furthermore, the proposed UML model provides a more exhaustive formalization than the architectures detailed in [11], [14]. Explicitly incorporating instantaneous feedback and contextual adjustment processes, often neglected in previous research. This approach offers us the opportunity to address still unresolved issues in the field, such as the administration of changes between various levels of complexity and the harmony between system direction and student independence.

II. Theoretical Framework

The theoretical framework of our work on the design and modeling of a dynamic adaptive hypermedia system is based on several key areas that include learning theories, system adaptability, hypermedia technologies, and static and dynamic interactive system modeling approaches. This framework is based on theories and concepts from artificial intelligence, software engineering, interface customization, and pedagogy.

A. *Learning Theories and Adaptive Systems*

Learning theories are at the heart of adaptive systems, as they influence how content should be structured and presented. They play a central role in the design of adaptive learning systems, especially in hypermedia environments. They provide the conceptual foundations that guide the adaptation of learning paths according to learners' profiles and needs. These theories explain how individuals acquire, process, and retain information, which helps to structure and personalize learning experiences. Learning theories include cognitive, constructivist, and behaviorist approaches, each with a specific impact on the design of adaptive learning environments [11], [23], [24]. Some of the most influential theories include:

1) Multimodal and adaptive learning: Multimodal learning theory proposes that learners process information more efficiently when it is presented in multiple forms, such as text, images, videos, interactive simulations, etc. This approach takes into account the fact that individuals have diverse preferences in terms of sensory modalities (visual, auditory, kinesthetic), which directly influences the design of adaptive learning systems. This theory makes it possible to design learning environments that can adapt to different learning styles [25], [26]. Adaptive hypermedia systems exploit this theory by offering a variety of teaching resources and adjusting content according to individual learners' preferences.

2) Cognitive and constructivist theories of learning: Cognitive and constructivist theories of learning provide fundamental foundations for the design of adaptive hypermedia systems. These theories share a common vision that learning is an active process where learners construct their own knowledge by integrating new information into their existing cognitive structures.

Indeed, cognitive theories focus on how learners perceive, process, and store information. This theory suggests that learners organize information in their long-term memory through processes such as encoding and recall [27], [28]. Adaptive hypermedia systems incorporate these principles by structuring content to avoid cognitive overload and by offering contextual help or additional explanations based on learners' answers or mistakes [11].

Constructivist theories, such as those of Piaget and Vygotsky, emphasize the active role of the learner in the construction of knowledge that allows him to build his own understanding of the world according to his experiences [23]. In an adaptive system, this implies that content must adjust according to the learner's skill level and learning style, as does modular content, which adjusts to the user's choices to meet their immediate interests or needs. By allowing the learner to actively explore resources, often in interactive environments, such as immersive environments or simulations, where the learner can manipulate variables and observe the results [11].

Cognitive and constructivist theories strongly influence the design of adaptive hypermedia systems by directing their ability to provide personalized, interactive, and contextual learning. These theories help to understand learners' needs and design tools that can dynamically adapt to their progress and preferences, enhancing the effectiveness of learning in modern digital environments.

3) Socioconstructivist learning and adaptation: Socioconstructivist theories emphasize the crucial role of social interactions in the learning process. These theories, inspired mainly by the works of Vygotsky and other researchers, postulate that learning is a social and contextual activity where individuals construct their knowledge through their interactions with others, cultural tools, and their environment [29], [23] . In the context of adaptive learning systems, these principles translate into the integration of interfaces, collaborative tools, and coping mechanisms that promote these interactions [11], [30].

4) Theory of Experience-Based Learning (Learning by Doing): The idea that individuals learn best by practicing and experiencing real-life situations is at the heart of modern approaches to learning and provides a solid basis for the design of adaptive learning systems [31], [32]. This perspective draws on several theories and pedagogical frameworks, including experiential learning, situated cognition, and constructionism. Indeed:

Kolb's theory of experiential learning proposes that learning occurs through a cycle of concrete experiences followed by reflections and applications [33]. Adaptive systems can leverage this theory to provide immersive learning experiences tailored to learners' needs. Adaptive systems use this cycle to provide interactive activities, simulations, and assessments that promote active learning.

Situated cognition suggests that learning is most effective when it takes place in contexts close to those where the knowledge will be applied [30]. Adaptive learning environments draw inspiration from this theory to create realistic scenarios, immersive tasks, or serious games where users can directly apply the skills.

According to the theory of constructionism, individuals learn by creating concrete artifacts [34]**.** Adaptive systems allow learners to build digital projects while receiving real-time feedback to adjust their actions.

The integration of real-life practices and experiences into adaptive systems is based on a sound theoretical foundation and offers an effective approach to meeting the varied needs of learners. By combining simulation, personalization and feedback, these systems promote active and relevant learning, making users better prepared to apply their knowledge in real-world contexts.

B. Adaptive Systems and Hypermedia

1) Concept and applications: Adaptive systems in education aim to adjust learning paths according to the specific characteristics of learners, such as their level of competence, learning preferences, or learning pace. [35], [36], [37], [38]. Hypermedia systems refer to non-linear information systems where users can navigate between multimedia resources, often interactively. These systems offer great flexibility and a richer learning experience compared to traditional teaching materials (books or lectures) [39], [40], [41], [42], [43].

When the two systems are combined, they leverage digital media (text, image, video, sound, etc.) to provide user-friendly educational resources and materials. Their integration helps create powerful learning environments that provide a personalized, multi-modal experience. In these environments, the educational content is not only adapted according to the learner's profile, but is also interactive and immersive, thanks to the use of multimedia technologies. Adaptive systems seek to meet the individual needs of users by adjusting resources based on their behavior, context, and preferences. These systems are often used in e-learning, content recommendation, and knowledge management applications. Indeed, adaptive hypermedia systems are based on the idea that the user can navigate through different types of content (text, video, image, etc.), with the possibility that each resource is adjusted according to the user. Such a system must incorporate the ability to analyze user interaction and adjust content or navigation paths in real time [44], [45], [11], [46], [47].

2) Characteristics of adaptive systems: An adaptive system is a system that is able to modify its behavior or responses based on user interactions or characteristics. In an educational context, this includes several dimensions [48], [49], [50], [51].

Performance-based adaptation: The system adjusts content based on the learner's previous actions, such as right or wrong answers in a quiz, or how quickly they complete a task.

Preference-based adaptation: Some adaptive systems adjust the path based on a learner's learning styles (visual, auditory, kinesthetic), interests, or priorities;

Context-based adaptation: The system can adjust its behavior according to the learning context, which could include the time of day, the environment, or the device used (computer, tablet, mobile phone).

These systems use adaptation algorithms, learner profiling models, and feedback mechanisms to deliver a personalized journey.

III. Results

1) Unified Modeling Language (UML): Dynamic adaptive hypermedia systems (SHAD) meet the diverse needs of users by adjusting their content, navigation, and presentation based on criteria such as preferences, context, and behavior. UML modeling is a standardized and powerful approach to representing, designing, and documenting software (and sometimes non-software) systems using graphical diagrams [51]. UML makes it possible to visualize the structure, behavior and interactions of the different components of a system, thus facilitating communication between stakeholders, both technical and non-technical. UML is widely used in software engineering to analyze functional and non-functional requirements, design robust software architectures, and document existing or developing systems [52], [53], [54], [55], [56]. Through this study, we propose four diagrams, including two structural diagrams (class diagram and use case diagram) and two behavioral diagrams (sequence diagram and activity diagram).

2) The class diagram: The class diagram is an essential tool for modeling the static structure of software systems. By clearly defining classes, their attributes, methods, and relationships, it plays a central role in object-oriented design. This diagram is particularly useful in educational contexts, where it allows to model complex learning environments, integrating various roles (teachers, learners, courses, content, etc.). Our diagram has eleven classes.

3) The Use case diagram: The use case diagram is an essential tool for understanding and modeling the interactions between a system and its users. Its simplicity and expressiveness make it a powerful communication tool to identify key features of the system while ensuring that they meet the needs of the stakeholders

4) The activity diagram: The activity diagram is a graphical representation used in the UML language to model workflows or processes in a system. It highlights the sequence of activities, decisions, bifurcations, and synchronizations in a given process.

5) The sequence diagram: The sequence diagram is an essential tool for modeling dynamic scenarios in a system. By showing how actors and components interact over time, it makes it possible to design, analyze and optimize complex processes. This methodology is particularly useful in educational contexts, where user-system interaction is crucial to provide a personalized and effective experience.

IV. Discussion

In this section, we discuss the modeling results in the form of a description of the various proposed diagrams:

A. The Class Diagram

The Fig. 1 presents the class diagram. It includes eleven interconnected classes, each playing a key role in meeting the educational needs of learners, teachers, and administrators, in the following format:

The LearningEnvironment class: This class is the nerve center of the learning environment system, managing the entire learning process. It has a unique identifier (id), is associated with a specific course, and stores the learner's environment preferences. The LearningEnvironment class uses these preferences to configure itself, recommend activities based on the learner's learning style (getActivityRecommendations), and evaluate the suitability of the current environment (evaluateEnvironment). For example, assessment Environment (preferences: Environment Preferences): bool: evaluates the current learning environment based on the learner's preferences and returns a Boolean value indicating whether the environment is suitable. These operations ensure that the learning experience is tailored to the needs of each learner, making LearningEnvironment the "brain" of the system that coordinates all interactions.

The LearningStyleProfile class: This class captures a learner's preferred learning style, allowing for personalized learning experience. It has a unique identifier (id) and is associated with a specific learner. The profile stores the learner's preferred learning styles (e.g., "Visual Learner", "Auditory Learner") and their preferred learning environment settings (environmentPreferences). For example, if a learner prefers a custom environment, this information will be stored in the EnvironmentPreferences attribute.

The LearningStyleProfile can be updated with new learner data (updateLearningStyles) and can determine the optimal learning environment for the learner based on their preferences (getOptimalEnvironment). This allows LearningEnvironment to tailor the learning experience to the individual needs of the learner, ensuring a more effective and engaging learning environment.

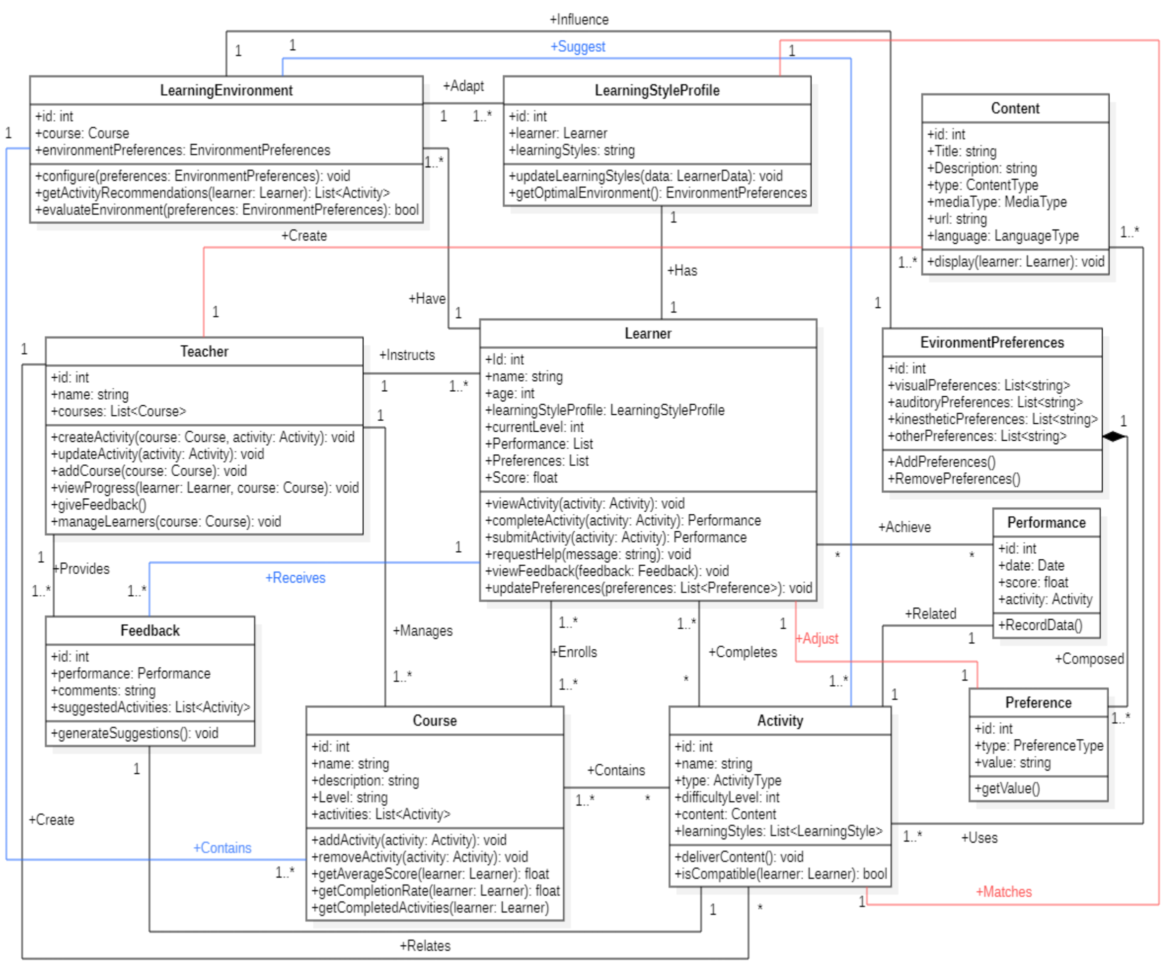

Fig. 1. Class Diagram.

Content Class: This class represents learning materials, including text, videos, images, quizzes, and other resources. The content provides the basic learning materials that learners interact with to acquire knowledge and skills. In this class, to define the different types of learning content, media formats, and languages in your system, we had to use enumerations. For example, the ContentType enumeration lists the different types of content such as "Video," "Text," and "Quiz." Similarly, the MediaType enumeration defines media formats such as "Video/mp4" and "text/html," while the LanguageType enumeration lists supported languages such as "English," "Spanish," and "French." Using enumerations for these attributes ensures that the values are consistent, readable, and easy to manage. Using enumerations makes the code more readable by providing meaningful names for the values instead of using raw strings. Enumerations also prevent invalid attributes from being accidentally assigned invalid values. Also, if we need to add new content types, media types, or languages, we only need to update the enumeration, not all the places where we use those values. Finally, you don't need to create another class with these names. For operations. Learner Void displays the content item to the learner.

The Learner class: This class represents the user of the learning environment system, the individual who learns and interacts with the system's features. The Learner can view the activities (viewActivity), complete them (completeActivity), submit them for grading (submitActivity), ask the teacher for help (requestHelp), and view the feedback provided by the teacher (viewFeedback). They can also update their preferences for the learning environment (updatePreferences). For example, a learner can complete an activity (completeActivity), receive feedback on their performance (viewFeedback), and then update their preferences (updatePreferences). The Learner is the central actor in the learning environment, engaging in activities, receiving feedback and monitoring their progress.

The EnvironmentPreferences class: This class captures a learner's preferences related to their learning styles, allowing them to customize their experience. It has a unique identifier (id) and stores a list of preferences categorized by learning style: visual (visualPreferences), auditory (auditoryPreferences), kinesthetic (kinestheticPreferences), and other (otherPreferences). For example, a learner may have visual preferences for learning via diagrams and videos, auditory preferences for listening to lectures and podcasts, and kinesthetic preferences for hands-on activities. The EnvironmentPreferences class allows you to add new preferences (addPreference) and remove existing ones (removePreference). This flexibility allows learners to adjust their preferences as needed, ensuring a personalized learning experience that aligns with their individual learning styles.

Performance class: This class records a learner's performance on a specific activity, by entering their score and the date the activity was completed. It also retains a link to the activity itself. For example, a Performance record might show that a learner scored 85% on an activity on a specific date. This class allows the system to track the learner's progress, identify areas where they may need additional support or guidance, and provide personalized feedback.

Activity class: This class represents a specific task or learning activity that learners participate in, such as taking a quiz, watching a video, or participating in a discussion. It has a unique identifier (id), a name, an ActivityType, a difficulty level, and is associated with a specific Content object. The activity also maintains a list of learning styles that it is compatible with (learningStyles). For example, a video activity may be compatible with visual learners, while a hands-on activity may be compatible with kinesthetic learners. The Activity class provides operations to deliver its content to the learner (deliverContent) and to check if it is compatible with the learner's learning style (isCompatible). This allows the system to recommend activities that are relevant and engaging for each learner, based on their individual preferences. The Activity class provides the building blocks of the learning experience, providing structured learning tasks that learners can perform to gain knowledge and skills.

Preference class: This class represents a specific preference related to a learner's learning style, allowing them to personalize their learning experience. It has a unique identifier (id), a type (PreferenceType), and a value (value). For example, a Preference object can represent a learner's preference for learning through visual aids (e.g., diagrams, videos), hearing aids (e.g., lectures, podcasts), or kinesthetic activities (e.g., hands-on projects). For the operation, getValue() retrieves the value of the preference. This would allow the system to access the preference information and use it to tailor the learning experience to the learner's individual learning style.

The Teacher Class: This class can create new activities for a specific course (createActivity), update existing activities (updateActivity), and add new courses to its list (addCourse). It can also view a specific learner's progress in a course (viewProgress), provide feedback to learners (giveFeedback), and manage learners enrolled in a specific course (manageLearners). For example, an instructor can create a new quiz activity for a specific course (createActivity), view a learner's progress in that course (viewProgress), and then provide feedback on their quiz performance (giveFeedback). The teacher plays a crucial role in guiding and supporting learners, providing instruction, feedback, and guidance to help them achieve their learning goals.

The Coursework class: This class represents a structured set of learning materials and activities that learners can enroll in to gain knowledge and skills in a specific subject area. It has a unique identifier (id), a name, a description, a level (for example, "Beginner", "Intermediate", "Advanced") and a list of activities included in the course (activities). The Course class allows you to add new activities (addActivity) and delete existing activities (remove Activity). It also provides operations to calculate a learner's average score in the course (getAverageScore), calculate a learner's completion rate in the course (getCompletionRate), and retrieve a list of activities completed by a learner in the course (getCompletedActivities). For example, an instructor can add a new activity to a course (addActivity), view a learner's average score in that course (getAverageScore), and then provide feedback on their performance based on their progress. The Courses class provides a framework for organizing and delivering learning content, allowing learners to focus on specific areas of study and track their progress.

The Feedback class: This class represents the feedback provided to a learner by a teacher about their performance in an activity. It has a unique identifier (id), is associated with a specific performance record, and stores feedback provided by the instructor (feedback). It also keeps a list of suggested activities based on the learner's performance (suggestedActivities). The Feedback class provides an operation to generate activity suggestions based on the learner's performance (generateSuggestions). This allows the teacher to provide personalized recommendations for further learning, helping learners improve their understanding and address areas where they may need additional support.

As a summary of our proposed classroom diagram, it includes eleven interconnected classrooms, each of which plays a key role in meeting the pedagogical needs of learners, teachers, and administrators. We propose a synthesis of the main classes and their interactions:

a) Learner-Centered Classes:

- Learner Profile and LearningStyleProfile: Capture individual learning preferences and styles to personalize learning experiences;
- EnvironmentPreferences: Allows the configuration of specific environments adapted to sensory preferences (visual, auditory, kinesthetic);
- Performance: Tracks the learner's progress and records scores for completed activities.

b) Content and activity classes:

- Content: Represents multimedia learning materials;
- Activity: Defines the instructional tasks associated with the content, compatible with different learning styles;
- Course: Structures content and activities into coherent modules for learners.

c) Interaction and feedback classes:

- Feedback: Provides personalized feedback and suggestions to improve performance;
- Teacher: Allows teachers to create and administer courses, provide feedback, and track learners' progress.

d) Central Class:

- LearningEnvironment: Coordinates all interactions between classes, ensuring maximum customization and efficiency.

In conclusion, the proposed class diagram illustrates a well-defined architecture for a digital learning system, highlighting:

Personalization: With classes like LearningStyleProfile and EnvironmentPreferences, the educational experience is tailored to the specific needs of learners;

Monitoring and feedback: The system promotes scalable learning, where performance is evaluated and used to generate targeted recommendations;

Flexibility and extensibility: The use of enumerations in classes such as Content and Activity ensures that they are easy to update and maintain.

B. The Case Diagram

In our case, the use case diagram, Fig. 2 depicts the dynamic interactions within a learning environment system, highlighting the roles of the teacher and learner.

The teacher, acting as a course creator, has the ability to design courses, develop individual activities, update existing activities, monitor the learner's progress, and provide feedback on completed tasks.

The learner, in turn, can register for courses, access and view course content, complete assigned activities, submit work for evaluation, ask the teacher for assistance, view feedback, personalize their learning experience by adjusting their preferences, and modify certain aspects of their learning environment.

The system itself plays a critical role in facilitating this interaction by recommending activities based on the learner's progress and preferences, analyzing the learning environment to identify areas for improvement, and dynamically adapting the learner's learning style based on their performance and interactivity, creating a personalized and engaging learning experience.

As a summary of our use case diagram, it models the interactions between the main actors (teachers, learners and system) within a personalized learning environment. It highlights the features and dynamic relationships needed to deliver an enriched and interactive educational experience. We mention the different roles of the main actors

a) The teacher:

- Content Creator and Administrator : Ability to create courses, design instructional activities, and update existing ones;
- Teacher Guide: Can monitor learners' progress, provide personalized feedback, and recommend appropriate activities.

b) The learner:

- Active Participant : Can view courses, complete assigned activities, and submit assignments for evaluation;
- Personalization: Can adjust learning preferences, ask for help, and view feedback to improve their experience.

c) The system:

- Interaction Facilitator : Recommends activities based on learners' progress and preferences;
- Dynamic Adaptation: Adjusts the learning environment and style based on learners' performance, providing continuous customization.

In conclusion, the diagram illustrates a balanced interaction architecture oriented towards personalized learning. Key features include:

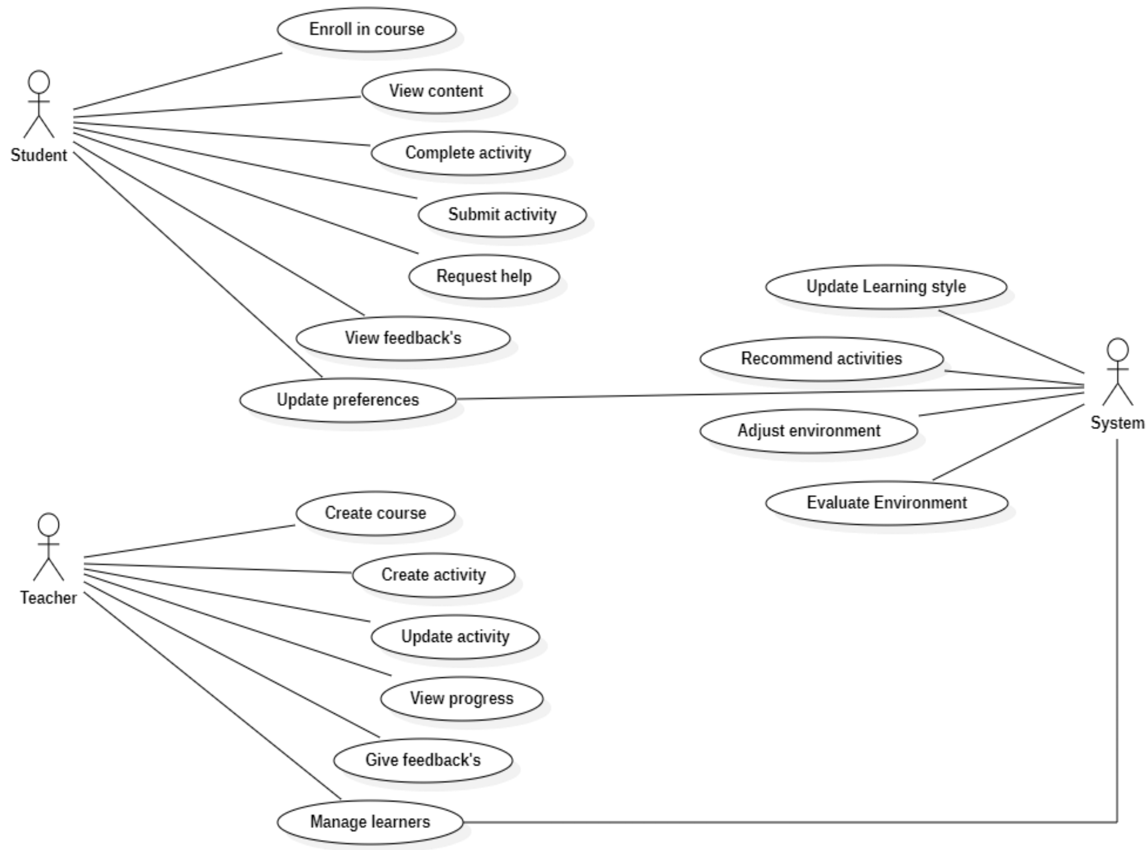

Fig. 2. Use Case Diagram.

Optimize learning with intelligent recommendations and dynamic adjustments.

Enhance learner engagement by providing an interactive and personalized experience.

Support the teacher in his or her role as a pedagogical guide with appropriate tools.

C. The Activity Diagram

The Fig. 3 presents the process, which begins by retrieving the learner's learning style and preferences, which are then used to configure the learning environment according to the learner's needs. Once the environment is set up, the system provides the relevant content to the learner and recommends appropriate activities based on their learning preferences. The learner then completes the recommended activity and their performance is recorded by the system, the synchronization relationship: the "Complete Activity" and "Record Performance" activities are synchronized, which means that both must be completed before moving on to the next step where the teacher provides feedback to the learner based on their performance. Then, the system analyzes its interactions with the learning materials. Based on this analysis, the learner's learning style is updated, and the environment can be reconfigured to better meet their changing needs. The system then recommends new activities that are tailored to the learner's updated learning style and preferences, starting the cycle all over again. This iterative process ensures that the learning experience is continuously tailored to the individual needs of the learner, providing a personalized and effective learning environment.

In summary, the proposed activity diagram models a dynamic and iterative process intended to personalize learning experiences within an education system. The key steps in this process highlight the use of learner preferences and performance to continuously adapt content and activities. Key steps include:

Learning Preferences and Style Collection: The system starts by retrieving data about the learner's learning style and preferences.

Configuration of the learning environment: This data allows you to configure an environment adapted to the specific needs of the learner.

Activity recommendation: The system proposes relevant educational activities based on the preferences collected.

Performance tracking and recording: The learner's performance is recorded after each activity, allowing for objective evaluation.

Personalized feedback: The teacher provides results-based feedback, improving human-system interaction.

Interaction analysis: The system analyzes the learner's interactions with the content to adjust future recommendations.

Learning style update: Based on the data collected, the learning style is reviewed, and the environment is reconfigured to better meet the evolving needs of the learner.

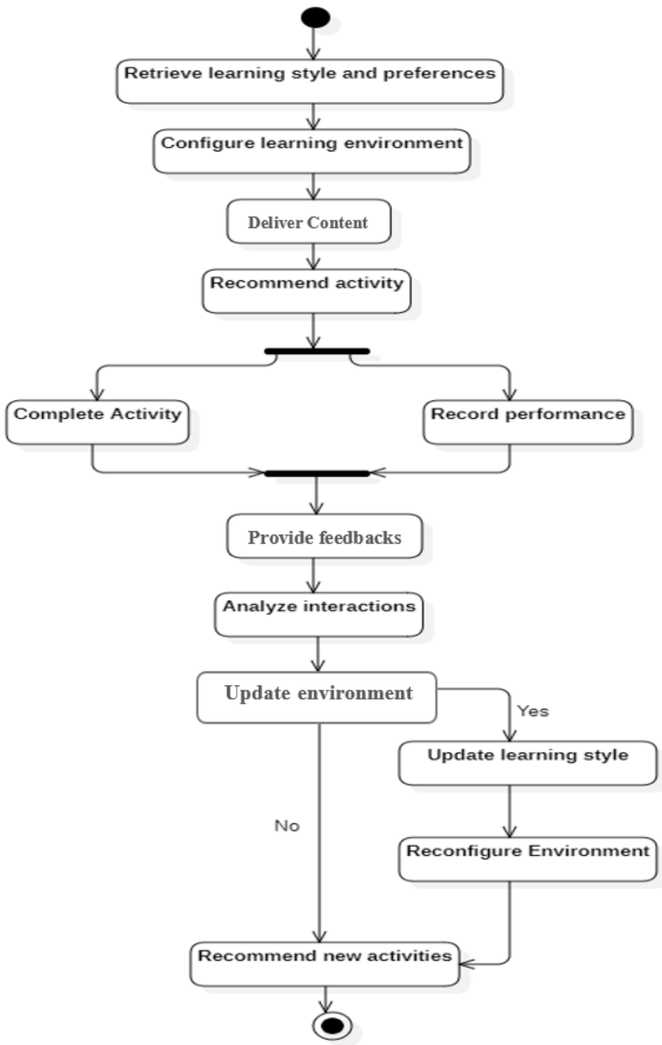

Fig. 3. Activity diagram.

This iterative process is designed to ensure continuous personalization, where each cycle improves the efficiency and relevance of the educational experience.

In conclusion, the activity diagram presented demonstrates an effective design for a personalized education system. Its strengths include:

Adaptability: Constant analysis of performance and interactions allows recommendations to be adjusted in real time.

Dynamic personalization: The iterative cycle ensures that the learning environment evolves with the needs of the learner.

The central role of the teacher: By providing qualitative feedback, the teacher plays a key role in optimizing the process.

D. The Sequence Diagram

The Fig. 4 presents the sequence diagram, which illustrates the interaction between the Teacher, System, Learner, Course, Activity, Learning Environment and Performance objects.

The teacher starts by creating a new course and several activities in the system. The system stores these objects. The learner then interacts with the system, potentially accessing the course and engaging in the activities. When the learner interacts with the activities, their performance is recorded in the Performance object. The system can then analyze the learner's performance and potentially trigger an update to the learning environment, depending on the analysis. This update may involve adjustments to the learning environment or recommending new activities based on the learner's performance.

It is important to note that the teacher can also access the performance data and provide feedback to the learner through the system. This feedback can be in the form of comments, suggestions, or advice.

The diagram highlights the collaborative nature of the learning environment, where the teacher creates and manages the content, the learner interacts with the system, and the system dynamically adapts to the learner's progress and needs, with the teacher providing feedback to improve the learning process.

In summary, the sequence diagram provided highlights the dynamic interactions between the key actors and components of an education system, such as the teacher, the learner, the system, the courses, the activities, and the learning environment. This model illustrates a collaborative process:

Role of the teacher: Creation of educational content and consultation of learners' performance data.

Learner role: Interaction with the system through the proposed activities, generating performance data.

System management: Data recording, performance analysis, dynamic adaptation of the learning environment, recommendation and new activities.

Personalized adaptation: Adjustments made to the learning environment based on learners' performance, enhancing pedagogical effectiveness.

The model also emphasizes the importance of feedback from teaching, stopping learning through guidance and complementary adjustments.

In conclusion, this sequence diagram illustrates an interactive educational and adaptive ecosystem, where the teaching, the learner, and the ecosystem system for the goals of the goals optimized. With a focus on personalization and dynamic feedback, it demonstrates how an environment of environment designed within the framework of individual learners' needs while maintaining an active role for teaching.

This modeled approach is particularly relevant in contexts where digital learning requires flexibility, responsiveness and collaboration to deliver a rich and engaging experience.

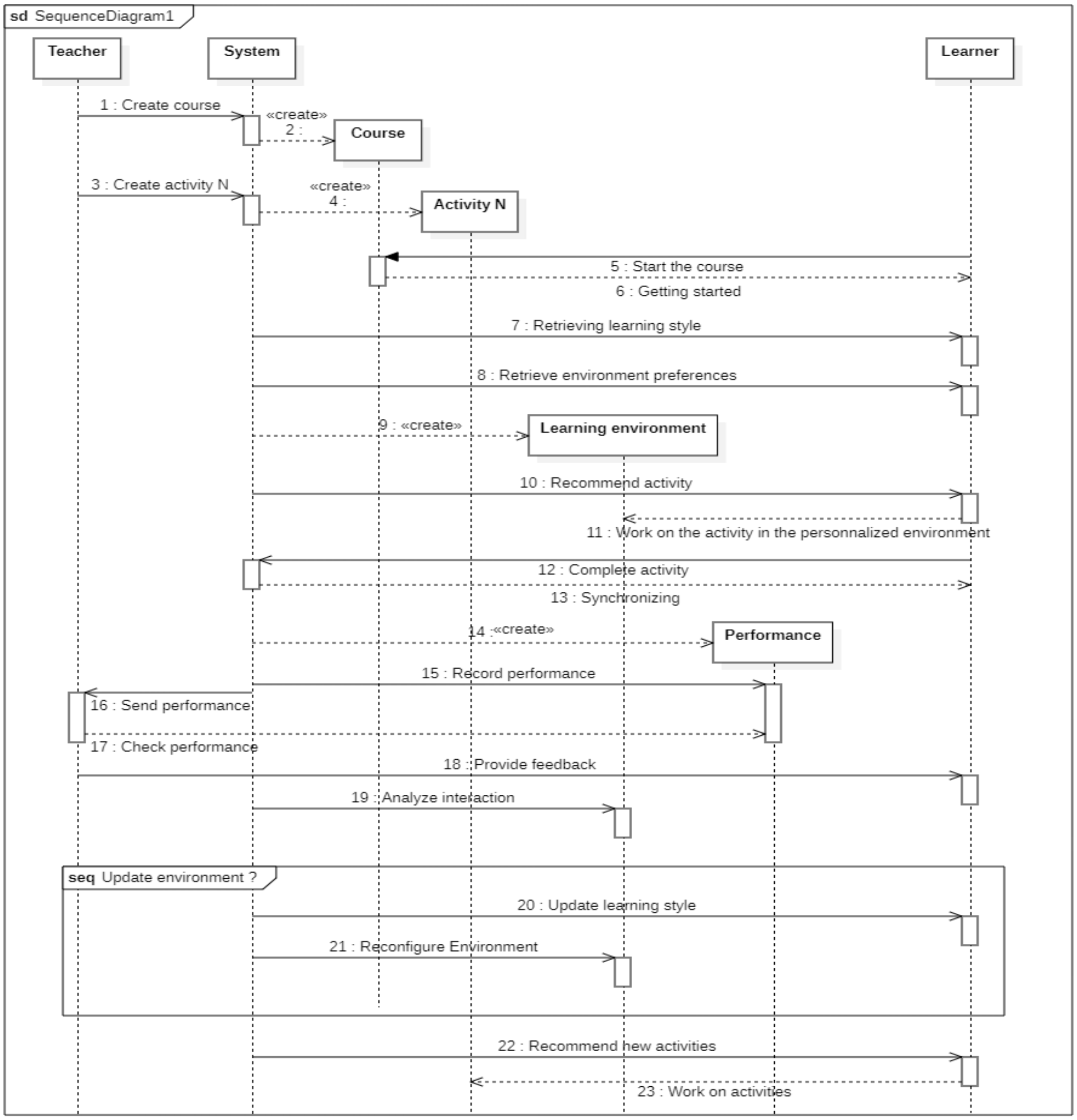

Fig. 4. Sequence Diagram.

V. Theoretical and Practical Contributions

This research makes three main contributions to the field of adaptive hypermedia systems. On a theoretical level, it proposes an innovative conceptual model that unifies content and user interface adaptation, thus filling a gap identified in the literature [57]. The developed UML modeling framework offers a more comprehensive formalization than existing architectures [58], explicitly integrating dynamic decision-making and contextual adjustment mechanisms. From a methodological perspective, our approach demonstrates how modeling techniques can be combined with AI algorithms to create more responsive and personalized learning systems. On a practical level, this work offers concrete benefits for various stakeholders in the educational field. For learners, the system offers a more engaging experience better adapted to their individual needs, which could improve retention and success rates. For instructional designers, our model provides a structured framework for developing adaptive content without requiring advanced technical skills. Educational institutions will find a scalable solution that can be gradually integrated into their existing infrastructures, with potential implications in terms of cost reduction and optimization of educational resources.

VI. Research Limitations

Several limitations deserve to be highlighted in this study. First, the current model relies on assumptions regarding the availability and quality of training data, which may not always be verified in real-world educational contexts. Second, although our UML approach allows for a comprehensive representation of the system, its practical implementation would require computational resources that could pose challenges in some capacity-limited environments. Another important limitation concerns the generalizability of the results: our preliminary validation, while promising, was conducted in a specific context and should be extended to more diverse learner populations. Finally, the current system does not yet fully address aspects related to interoperability with other learning platforms, a crucial dimension for widespread adoption. However, these limitations open up interesting avenues for future research, particularly on the optimization of adaptation algorithms and integration with existing educational standards.

VII. Suggestions for Future Research

Although the technologies and architectures of adaptive systems have evolved enormously, several challenges remain:

1) Next, we will develop the prototyping of the system based on the diagrams developed in this study, and ultimately develop our hypermedia system.

2) Current adaptive systems often use AI models, which limits teachers' and learners' confidence in the proposed recommendations. We can then develop explainability mechanisms (XAI) integrated into UML diagrams (annotations in sequence diagrams to trace adaptation decisions).

3) Current profiles often ignore emotional states (frustration, motivation), which are critical in pedagogy. We can therefore extend the class diagram with emotional attributes (biometric data or textual analyses).

VIII. Conclusion

Adaptive learning systems represent a major advancement in digital pedagogy, drawing on a synergy between artificial intelligence, data analytics, and cloud infrastructures. These technologies enable the delivery of truly personalized experiences, where every element—content, pace, assessment methods, and interface—dynamically adjusts to the learner's needs. Modern microservice architectures offer the flexibility to continually integrate new capabilities, such as emotion analysis or augmented reality, while ensuring performance and security. However, this potential comes with crucial challenges: algorithm transparency, energy footprint reduction, and sensitive data protection. The future of these systems lies in their ability to combine technological sophistication with a human-centered approach, while adhering to open standards for widespread adoption. The next frontier will be developing predictive and immersive learning ecosystems that can not only adapt to learners, but also anticipate their needs throughout their educational journey.

References

[1] Nanard, M. "Hypertexts: Beyond Links, Knowledge". Educational Sciences and Techniques (STE)., vol 2, 1. pp 31-59. 1995

[2] BRUILLARD, E (1997). Teaching machines. Paris: Hermès.

[3] Delestre, N. (2000). Metadyne, a dynamic adaptive hypermedia for teaching (Doctoral dissertation, University of Rouen).

[4] Kostadinov, D. (2003). Personalization of information and management of user profiles. Master's thesis PRiSM, Versailles.

[5] Hofmann, M., & Pustokhina, I. Personalization and Recommendation in E-Learning. In Recommender Systems Handbook (pp. 681-707). Springer.2017

[6] Moura, F., & Moreira, A. Personalized Learning Path Generation for E-learning Systems. International Journal of Emerging Technologies in Learning (iJET), 12(8), 4-14.2017

[7] Alammary, A., Sheard, J., & Carbone, A. (2014). Blended learning in higher education: The students' learning experience. European Journal of Open, Distance and E-Learning.

[8] Siemens, G. Learning analytics: The emergence of a discipline. American Behavioral Scientist, 57(10), 1371-1381.2013

[9] Drachsler, H., & Koper, R. Personalized learning and the role of technology. Journal of Educational Technology & Society, 13(3), 26-39.2010

[10] Baker, R. S. J. d., & Yacef, K. The state of educational data mining in 2009: A review and future visions. Journal of Educational Data Mining, 1(1), 3-17.2009

[11] Brusilovsky, P., & Millán, E. User models for adaptive hypermedia and adaptive educational systems. In The Adaptive Web (pp. 3-53). Springer.2007

[12] Kobsa, A. User Modeling: Recent Work, Prospects and Challenges. User Modeling and User-Adapted Interaction, 11(1-2), 49-78.2001

[13] Höfer, T., & Hesse, F. W. (2004). Adaptive Learning Environments: A User-Centered Approach. Educational Technology & Society.

[14] Brusilovsky, P. (2001). Adaptive Educational Systems and Educational Hypermedia: From Design to Implementation. In Hypermedia and Intelligent Tutoring Systems: Advanced Applications.

[15] Meyer, M., & Freitas, D. (2016). Modeling Adaptive Hypermedia Systems with UML. Software Engineering Research, Management and Applications.

[16] Sommerville, I. (2011). Software Engineering (9th edition). Addison-Wesley.

[17] Koedinger, K. R., & Corbett, A. T. Cognitive tutors: From the lab to the classroom. AI Magazine, 27(4), 5-19.2006

[18] Ambler, S. W. (2004). The Object Primer: Agile Model-Driven Development with UML 2.0. Cambridge University Press.

[19] Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley.

[20] Larman, C. (2004). Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. Prentice Hall.

[21] Bertolino, A., & Gotsman, A. Modeling and verification of dynamic adaptive systems. In International Conference on Formal Engineering Methods (pp. 99-115).2004 Springer.

[22] Osterweil, L. J., & Schneider, G. Modeling dynamic systems with UML: A practical approach. Software and Systems Modeling, 1(3), 275-289.2002

[23] Piaget, J. (1970). Science of Education and the Psychology of the Child. Viking Press.

[24] Fleming, N. D., & Mills, C. Not Another Inventory, Rather a Catalyst for Reflection. To Improve the Academy, 11(1), 137-155.1992

[25] Gardner, H. (1983). Frames of Mind: The Theory of Multiple Intelligences. Basic Books

[26] Sweller, J. Cognitive load during problem solving: Effects on learning. Cognitive Science, 12(2), 257-285.1988

[27] Ausubel, D. P. (1968). Educational Psychology: A Cognitive View. Holt, Rinehart, and Winston.

[28] Wenger, E. (1998). Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press.

[29] Brown, A. L., Collins, A., & Duguid, P. Situated Cognition and the Culture of Learning. Educational Researcher, 18(1), 32-42.1989

[30] Gee, J. P. (2003). What Video Games Have to Teach Us About Learning and Literacy. Palgrave Macmillan.

[31] Schank, R. v. Goal-Based Scenarios: A Radical Look at Education. The Journal of the Learning Sciences, 3(4), 429–453.1994

[32] Kolb, D. A. (1984). Experiential Learning: Experience as the Source of Learning and Development. Prentice Hall.

[33] Papert, S. (1980). Mindstorms: Children, Computers, and Powerful Ideas. Basic Books.

[34] Graesser, A. C., Conley, M. W., & Olney, A. (2012). Intelligent Tutoring Systems. In APA Educational Psychology Handbook.

[35] Fletcher, J. D., & Morrison, J. E. (2012). DARPA Digital Tutor: Assessment Study. Institute for Defense Analyses.

[36] Woolf, B. P. (2010). Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning. Morgan Kaufmann.

[37] Dagger, D., Wade, V. P., & Conlan, O. Personalisation for All: Making Adaptive Course Composition Easy. Educational Technology & Society, 8(3), 9-25.2005

[38] Schroeder, R. Being There Together: Social Interaction in Shared Virtual Environments. Presence: Teleoperators and Virtual Environments, 15(4), 1-13. (2006).

[39] Anderson, T., & Elloumi, F. (2004). Theory and Practice of Online Learning. Athabasca University Press.

[40] De Bra, P., & Calvi, L. AHA! .(1998) An Open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 4(1), 115-139

[41] Landow, G. P. (1997). Hypertext 2.0: The Convergence of Contemporary Critical Theory and Technology. Johns Hopkins University Press.

[42] Nielsen, J. (1995). Multimedia and Hypertext: The Internet and Beyond. Morgan Kaufmann.

[43] Sharda, R., Delen, D., & Turban, E. (2020). Business Intelligence, Analytics, and Data Science: A Managerial Perspective. Pearson.

[44] O'Reilly, M., & McNamara, M. The role of adaptive learning technologies in education. Educational Technology Research and Development, 67(3), 479–491.2019

[45] Cristea, A., & Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia. In Adaptive Hypermedia and Adaptive Web-Based Systems (pp. 122-132). 2002.Springer.

[46] Brusilovsky, P. Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 11(1-2), 87-110.2001

[47] Pérez-Marín, D., & Pascual-Nieto, I. (2011). Conversational Agents and Natural Language Interaction: Techniques and Effective Practices. IGI Global.

[48] VanLehn, K. The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197-221.2011.

[49] Woolf, B. P. (2010). Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning. Morgan Kaufmann.

[50] Shute, V. J., & Towle, B. Adaptive E-Learning. Educational Psychologist, 38(2), 105-114.2003

[51] OMG, (2022). Unified Modeling Language (UML) Specification.

[52] Dennis, A., Wixom, B. H., & Tegarden, D. (2020). Systems Analysis and Design: An Object-Oriented Approach with UML. Wiley.

[53] Müller, M., & Reichenbach, M. (2014). A model-driven approach to adaptive systems design using UML profiles. Proceedings of the ACM Symposium on Applied Computing.

[54] Pressman, R. S., & Maxim, B. R. (2014). Software Engineering: A Practitioner's Approach (8th Edition). McGraw-Hill.

[55] Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide (2nd Edition). Addison-Wesley.

[56] Fowler, M. (2004). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesle

[57] Kobsa, A. (2007). *Generic user modeling systems*. User Modeling and User-Adapted Interaction, 11(1-2), 49-63.

[58] Carmona, C., et al. (2008). Designing dynamic adaptive evaluation systems. IEEE TLT, 1(2), 73-85.