Modeling an Adaptive and Collaborative E-Learning System with Artificial Intelligence Tools

Kawtar Zargane*, Hassane Kemouss, Mohamed Khaldi
A Research Team in Computer Science and University Pedagogical Engineering-(S2IPU) Normal School of Tétouan, Abdel Malek Essaadi University, Morocco

***Abstract*—In an educational environment that is undergoing digital transformation, the need to create smarter, learner-centred learning environments is becoming increasingly urgent. This article presents a conceptual modeling of an e-learning system that integrates the adaptive and collaborative dimensions, relying on the tools of artificial intelligence (AI), which occupies a central place, both as a dynamic adaptation engine and as a collaboration facilitator and automaton of certain pedagogical activities. This methodical and structured approach makes it possible to develop a hybrid environment capable of adjusting to individual needs while promoting the co-construction of knowledge between peers. Based on instructional design principles and the 2TUP (Two Tracks Unified Process) process, this approach aims to develop a systematic architecture, illustrated by UML (Unified Modeling Language) diagrams, of classes, use cases, activities and sequences, integrating AI (Artificial Intelligence) through adaptive learning, conversational agents and intelligent tutoring systems that make it possible to personalize learning, Provide targeted feedback, optimize learner performance, and guide learners more accurately. This combination of standardized modeling and AI improves the synergy between stakeholders and increases the efficiency of online learning environments. Finally, this model paves the way for a new era of more flexible, inclusive and responsive techno-pedagogical systems, capable of facing the contemporary challenges of online training.**

Keywords*—Conceptual modeling of an online learning system; Adaptive system; Online collaborative learning; Artificial intelligence (AI); Educational software architecture; UML modeling

I. Introduction

In the digital age, in a context where online learning is constantly developing, and in the face of the diversity of learner profiles, the complexity of journeys and the growing importance of collaboration, it is becoming imperative to design smarter, more flexible and learner-centred learning systems, it is becoming essential to design educational experiences that are both effective, customized and technologically robust. This study highlights the importance of a rigorous approach in software engineering to design pedagogical devices capable of adapting to the needs of learners [1]. By combining instructional design principles with software development best practices, this allows us to develop a systematic and consistent architecture, modeled using UML diagrams, to effectively structure these e-learning projects. The integration of artificial intelligence (AI) into this architecture represents a powerful lever for the personalization of pathways, taking into account the pace, profile and specific difficulties of each learner [2,3].

In this perspective, the integration of collaborative learning highlights the social dynamics of knowledge construction, by promoting interaction between peers, the co-creation of resources and the development of transversal skills. At the same time, AI-supported adaptive learning makes it possible to personalize the educational experience by adjusting the content, pace and pedagogical strategies and to automate certain tasks, analyze learners' performance and offer precise pedagogical recommendations based on the specific characteristics of each learner [4, 5]. This combination can profoundly transform education [6]. Through the use of intelligent chatbots and adaptive tutoring systems, this innovative learning system is able to generate targeted feedback, detect the specific needs of each user and recommend personalized content [7]. This approach, which combines personalization [8], collaboration and artificial intelligence, is based on an intelligent tutor that adjusts content and activities in real time according to the learner's profile. The collaborative dimension is fully integrated through exchange spaces, online discussion tools, and group work functionalities [9]. Continuous monitoring of progress also allows for the adjustment of pathways as needed, ensuring a responsive, stimulating learning environment that is conducive to autonomy [10].

This article thus proposes a conceptual modeling of an online learning system that is both adaptive and collaborative, based on AI tools. The objective is to lay the foundations of an intelligent environment capable of reconciling individual personalization and collective intelligence, in a learner-centered, scalable and equitable learning perspective. To meet pedagogical and technical requirements, the adoption of a structured design approach based on the 2TUP methodology is necessary. This allowed the full potential of the UML modeling language to be fully exploited through two complementary perspectives: a static view, with class and use case diagrams describing the architecture of the system, and a dynamic view, illustrated by activity and sequence diagrams translating the interactions and behaviors of the system. These visual representations reinforce the overall understanding of the system and concretely demonstrate the articulation between modeling, AI and collaborative learning.

This model paves the way for a new generation of technopedagogical systems that are more flexible, inclusive, and responsive, capable of meeting the current challenges of online training.

II. Conceptual and Theoretical Framework

This conceptual and theoretical framework aims to inform the design of an e-learning system based on three complementary domains that intersect to structure the pedagogical, technological and methodological approach of the proposed system. The pedagogical foundation is based on collaboration between peers, inspired by the work of Vygotsky this approach considers that knowledge is built through social interaction, cooperation between peers and dialogue [11]. Then, the second theoretical pillar concerns artificial intelligence in education, and more specifically adaptive learning systems and intelligent agents. These technologies make it possible to personalise the courses according to the profile, pace and needs of each learner. Finally, the third axis is based on educational software engineering, which provides a rigorous methodology for designing complex systems. By articulating these three dimensions, this work is part of a systemic approach, laying the foundations for a fine modeling of an online learning environment that is both personalized, collaborative and intelligent, capable of meeting the current challenges of distance learning.

This theoretical framework therefore guides the technical design and architecture of the adaptive and collaborative learning system maintained by AI is proposed, detailed according to:

A. The Pedagogical Approach

1) Collaborative learning: Collaborative learning, based on the principles of social constructivism, is based on the idea that knowledge construction is optimized when it is done in interaction with others. According to Johnson and Johnson, the pillars of this approach include positive interdependence, individual responsibility, and learning-promoting interactions [12]. By placing cooperation at the heart of the pedagogical system, the learning environment must encourage active exchanges between peers, thus allowing the confrontation of ideas, the collective resolution of problems and the development of social skills,[13]. To support this dynamic, the system must integrate appropriate collaborative tools (discussion forums, wikis, or shared workspaces), while ensuring balanced group participation [14]. Beyond simple interaction, one of the major objectives of collaborative learning is the co-construction of knowledge, where learners produce resources or solutions with high added value together.

Scardamalia and Bereiter emphasize the importance of supporting the complex cognitive processes involved in this co-elaboration: collective regulation, integration of divergent points of view, and shared reflection. Learning systems must therefore offer explicit pedagogical scenarios, as well as mechanisms for mediation or assistance in structuring collaborative work. Features such as virtual whiteboards, collaborative editors, or lightweight project management tools can promote this co-construction [15].

Finally, the management of roles and contributions through traceability systems, participation indicators or recognition mechanisms (badges, comments, peer evaluations), is a crucial lever to guarantee equity, commitment and efficiency of group work. As Strijbos and De Laat, [16] point out, it can strengthen learners' motivation and encourage active participation [17]. By integrating these dimensions, a well-designed collaborative learning system becomes a real trigger for collective intelligence, in the online learning experience.

B. The Technological Approach

1) Adaptive learning: Adaptive learning is now one of the major levers for personalizing the online educational experience. It is a pedagogical approach that dynamically adjusts the content, pace, and activities based on each learner's needs, skill level, and learning style.[18]. This personalization is based on a fine modeling of the user, integrating variables such as prior knowledge, cognitive preferences or learning styles, as proposed by Felder and Silverman and discussed more recently by Pashler et al ,[19],[20]. The objective is to make learning more relevant, motivating and effective by adapting the course to each profile.

Intelligent Tutoring Systems (ITS), platforms such as DreamBox or Knewton, as well as adaptive digital environments, illustrate these principles. These systems are based on a continuous evaluation of the learner's progress, made possible through the real-time analysis of their interactions with the platform. Algorithms monitor performance trends and identify gaps, making it possible to recommend specific resources, modify the sequence of content, or propose targeted remediation activities [21], [22]. The formative approach, as advocated by Bloom [23], is implemented here through integrated assessments, oriented towards pedagogical adjustment.

A third essential pillar of adaptive learning is personalized feedback. Beyond simple correction, this feedback aims to support metacognitive regulation, helping the learner to become aware of mistakes, to understand poorly mastered concepts and to plan future efforts [24], [25]. It can take the form of contextual comments, detailed explanations, recommendations for additional activities, or tailored motivational messages. This feedback, based on a detailed analysis of the learning data, helps to strengthen engagement and autonomy.

Thus, an effective adaptive learning system is not limited to distributing content; it acts as an intelligent tutor, capable of observing, analyzing and intervening in a targeted manner, to maximize the learning potential of each individual [26]. Despite their potential, these systems still have limitations, particularly in terms of general learner modelling, algorithm and data transparency, all of which are challenges for a more reliable and inclusive integration of AI in education.

2) Artificial intelligence in education: Artificial intelligence (AI) is now central to modern online learning systems, not only as a technological tool, but as a vector for educational intelligence. Its integration is profoundly transforming the ways in which digital learning environments are adapted, collaborated and automated [27].

The first major contribution of AI lies in its ability to personalize the learning experience. Unlike traditional fixed-rules-based approaches, AI-enabled systems use advanced data processing and machine learning techniques to analyze, in real time, learner interactions (behaviors, response time, recurring errors, engagement levels). These analyses make it possible to dynamically adapt the content, the level of difficulty, the pace or the materials offered, according to the individual profile and the evolution of each learner [28]. This fine adaptability, based on learning traces, reinforces the relevance of the educational paths and promotes commitment, self-regulation and autonomous progression.

Beyond individual personalization, AI also facilitates collaboration between learners. Thanks to natural language processing and social learning network analysis technologies, it is able to observe interactions in forums, contributions in shared spaces or roles in collective activities [29]. It can thus detect imbalances, propose strategic groupings or intervene as an intelligent mediator, by structuring exchanges, relaunching discussions or formulating open questions promoting the co-construction of knowledge. AI thus contributes to a more efficient, equitable and reflective management of collaborative work, by strengthening inclusion and group dynamics.

A third fundamental axis is the automation of repetitive pedagogical tasks, both for teachers and learners. Intelligent systems now allow the automatic generation of personalized quizzes, the correction of complex exercises, the synthesis of discussions, or the early warning in the event of a risk of disengagement or failure [30, 31]. These features reduce the cognitive and temporal load on teachers, while maintaining a quality of individualized follow-up. They also allow for instantaneous, contextualized feedback, which is essential for self-regulated learning.

Finally, although the contributions of AI are promising, its deployment in education systems raises important technical challenges. Among the major challenges are the transparency of algorithms, the explain ability of automated decisions and the fairness in the proposed recommendations [32, 33, 34]. It therefore becomes crucial to design educational AI systems that are not only efficient, but also accountable, learner-centred and inclusive.

Finally, artificial intelligence is now an essential lever for pedagogical innovation. It acts as a catalyst that, far from replacing the learner, reinforces the adaptive, collaborative and formative dimensions of learning, while laying the foundations for a more intelligent, equitable and sustainable educational ecosystem [2],[4].

C. The Software Approach

UML (Unified Modeling Language) modeling allows the structure and behavior of the system to be represented, while the 2TUP (Two Track Unified Process) method offers an iterative and scalable framework for software development [1]. Combined with a reflection on instructional design, these approaches ensure consistency between instructional objectives and technological choices.

1) The development process: The 2TUP process an evolution of the unified process, offers a rigorous and flexible framework for the development of information systems [35]. By separating the functional aspects, related to business needs, from the technical aspects, related to the system architecture [36]. The 2TUP makes it possible to better adapt to the constant evolution of technological environments and user requirements. This two-pronged approach fosters close collaboration between development teams and business lines, improving the quality of the final product. In addition, by incorporating regular iterations and project reviews, the 2TUP reduces the risks associated with late changes and enables continuous delivery of value [37].

Unlike more rigid methods, 2TUP offers greater agility while ensuring a high level of quality. For example, in the development of mobile applications, 2TUP has demonstrated its ability to adapt quickly to changes in platforms and user feedback, while maintaining functional consistency [38].

2) The UML modeling language: UML (Unified Modeling Language), a universal language for object modeling, offers a range of graphical tools to accurately and concisely represent the structure and behavior of a system [39]. Whether designing a software application, a database, or a business process, UML provides a common language for developers, analysts, and users [40]. Thanks to its various diagrams (use cases, classes, sequence, activity, etc.), UML allows you to visualize the different facets of a system, from user interaction to implementation details. This graphical representation greatly facilitates understanding, communication and collaboration within a development team [41]. In addition, UML is an evolving language that adapts to new technologies and changing project needs. In short, UML is an essential asset for any professional wishing to design and develop quality software systems [42].

Ultimately, UML remains a central modeling tool, offering rigor, clarity and shared vision throughout the life cycle of a project. Its ability to represent both the functional and technical aspects of a system makes it an essential lever for the design of robust and sustainable software solutions.

D. Critical Analysis of Existing Systems

In order to design a more effective e-learning system, it is essential to analyze the limitations of the solutions currently available, [43,44]. This section critically examines existing adaptive and collaborative systems, highlighting their contributions, but also their weaknesses in personalization, social interaction, and the integration of artificial intelligence. These findings will help guide the modelling of the proposed system.

1) Comparison of existing adaptive and collaborative e-learning systems: In this section, a critical analysis of adaptive and collaborative e-learning systems is conducted in order to identify their strengths and limitations. Adaptive systems personalise learning pathways according to learners' individual needs [43] while collaborative systems promote social interaction to strengthen motivation and co-construction of knowledge [45], [46]. However, these approaches are often implemented in isolation, without any real integration between personalization and collaboration [42]. In addition, their effectiveness is sometimes limited by a lack of granularity in adaptation or weak orchestration of collective activities. This comparison aims to shed light on the design orientations of a more efficient technopedagogical system, harmoniously integrating artificial intelligence, individual adaptation and social collaboration [47].

TABLE I. COMPARISON OF EXISTING ADAPTIVE AND COLLABORATIVE E-LEARNING SYSTEMS

Aspect	Adaptive e-learning systems	Collaborative e-learning systems
Main Objective	Personalize the learning experience based on the individual characteristics of the learner.	Harness the collective intelligence and social benefits of group learning.
Key Mechanisms	Modeling of the learner, the domain, and pedagogical adaptation.- Adaptation of the content or course according to the prerequisites or performances.- Detection of gaps and proposal of remedies.	Interaction, sharing of ideas, co-construction of knowledge.- Communication and collaborative work tools.
Examples	ITS (Intelligent Tutoring Systems), Knewton (Alma), DreamBox Learning.	Forums, Wikis, EVA, LMS (Moodle, Canvas, Blackboard).
Bounds	Adaptation often superficial, based on rigid rules.- Difficulty in integrating complex aspects such as motivation or emotion.- Dependence on the quality and quantity of data collected.- Little or no integration of social dimensions.	Collaborative tools without intelligent facilitation (little intervention on group dynamics).- Difficulty in valuing and measuring individual contributions.- Little synergy with adaptive systems (operating in silos).

According to Table I, Adaptive learning systems, such as Knewton (now integrated into the Alma platform), DreamBox Learning or Intelligent Tutorial Systems (ITS), aim to personalize the learner's experience by dynamically adapting the content, difficulty and paths according to their performance and profiles [21]. However, these devices often have limited adaptation, based on static rules or simple statistical models, struggling to integrate more qualitative variables such as motivation or emotional state. At the same time, collaborative platforms such as Moodle, Canvas or forum environments offer tools for interaction (discussions, sharing, group work), but collaboration remains poorly guided: group dynamics, role coordination or the valuation of contributions are rarely handled in an intelligent way [13]. This double observation reveals a persistent compartmentalization between individualized adaptation and collaborative learning. Data from learner profiles are rarely used to form relevant groups or adapt social interactions, and there is no automated orchestration to synchronize adaptive and collaborative processes within the same system [48]. This separation hinders the emergence of truly intelligent learning environments, fully integrating the social and cognitive dimensions.

III. RESULTS AND DISCUSSION

A. UML Modeling of an Integrative Learning System

UML modeling is a powerful tool for clearly and consistently representing the essential components of a complex system. In this section, we present a functional modeling of an integrative e-learning system, combining adaptivity, collaboration, and artificial intelligence. The main objective is to describe, by means of UML diagrams, a modular architecture as highlighted by Ouelhadj et al (2020), as well as Chen et al. (2023), which is capable of personalizing learning paths, managing social interactions and automating certain pedagogical tasks thanks to AI [40], [28].

This modeling thus aims to lay the foundations of an intelligent system, centered on both the individual and the collective, while facilitating the analysis, design and implementation of an evolving and adaptable learning environment [33], [41].

Based on the 2TUP development model and using the UML language, our study relies on the StarUML tool to accurately model the complex interactions between the learner, the content and the artificial intelligence. This work makes it possible to design tailor-made learning paths, adapted to the specific needs of each user.

1) Static part: 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 [49],[50]. Through this article, we propose four diagrams, including two structural diagrams (class diagram and use case diagram) and two behavioral diagrams (sequence diagram and activity diagram).

a) Class diagram: For Rosenberger, M. (2022) and Chehida, S. et al. (2021), a class diagram is used to illustrate classes along with their attributes, methods, relationships (such as association, aggregation, composition, inheritance, etc.) and a constraint [39],[51].

b) Diagram interpretation: Fig. 1 presents the static structure of the system, it allows to understand how the objects interact, to visualize the organization of the data and to clearly define the responsibilities of each class in the overall functioning of the system, as follows:

- TutorIntelligent Class: The TutorIntelligent class manages personalized learner coaching via the evaluateLearnerLevel(), adaptContent(), generate Exercises(), provideFeedbackPersonalized (), and followProgress() operations. This central class coordinates the adaptation of learning
    - evaluateLearnerLevel (): Allows you to evaluate the learner's current level. For example, it can analyze the learner's answers to programming exercises to determine their level in Python.
    - adaptContent(): Allows you to adjust the learning content according to the learner's level and profile. For example, it can offer more complex exercises if the learner masters the basics well.

- generateExercises(): Allows you to create custom exercises. For example, it can generate math exercises adapted to the learner's current level.
- provideFeedbackPersonalized (): Allows you to give detailed feedback on the learner's responses. For example, it can explain why an answer is incorrect and suggest ways to improve it.
- followProgress(): Allows you to track the progress of learning. For example, it can identify concepts that require more practice.

* Learner Group Class: The Learner Group class manages groups of learners through the setObjectives(), planActivity(), assignTutor(), evaluateProgress(), and WorkOnTheTask() operations. It organises and coordinates collective learning.
  - setObjectives(): Allows you to set learning objectives for the group. For example, they can set common goals for a collaborative project.
  - planActivity(): Allows you to plan learning activities. For example, they can organize group work sessions or practical workshops.
  - assignTutor(): Allows you to assign a tutor to the group. For example, he can assign a specialized tutor depending on the needs of the group.
  - evaluateProgress(): Allows you to evaluate collective progress. For example, it can measure the group's progress against the set goals.
  - WorkOnTheTask(): Allows you to manage work on assigned tasks. For example, it can track the distribution and progress of tasks among members.

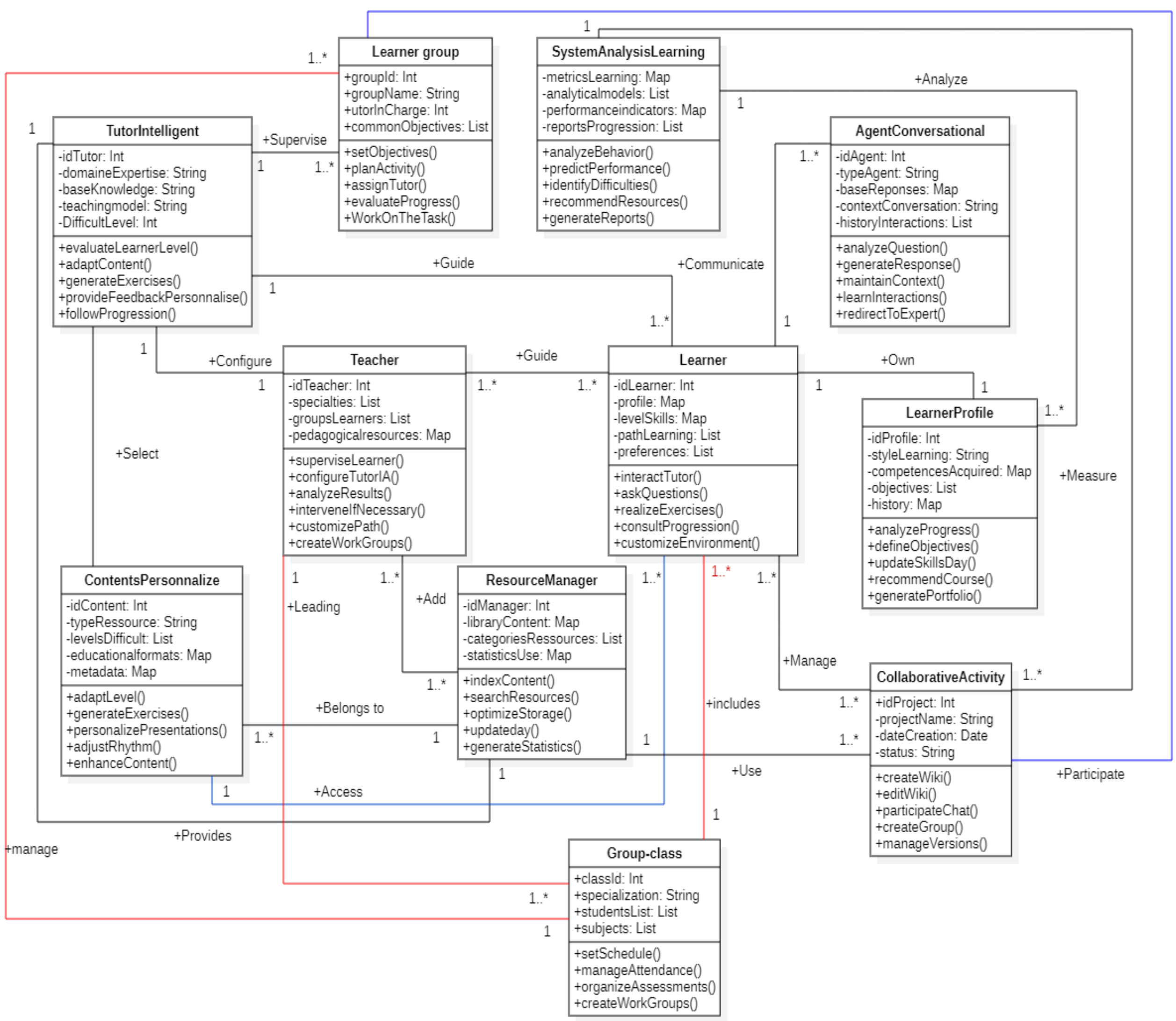

Fig. 1. Class diagram and static structure of online learning system.

- Teacher Class: The Teacher class oversees the learning process through the superviseLearning(), configureAITutor(), analyzeResults(), interveneIf Necessary(), and customizePath() operations. It represents the human expert who guides and adjusts the system.
  - superviseLearning(): Allows you to track the overall progress. For example, the teacher can observe the learning patterns of a group of learners.
  - configureAITutor(): Allows you to configure the tutorial system. For example, the teacher can adjust the evaluation criteria or difficulty thresholds.
  - analyzeResults(): Allows you to study learners' performance. For example, the teacher can identify concepts that cause problems for multiple learners.
  - interveneIfNecessary(): Allows you to provide human assistance. For example, the teacher can organize a remedial session for particularly difficult concepts.
  - customizePath(): Allows you to tailor learning paths. For example, the teacher can create specific learning paths for different learner profiles.
- Learner Class: The Learner class interacts with the system through the interactTutor(), askQuestions(), effectExercises(), consultProgress(), and customize Environment() operations. It represents the main user of the learning system.
  - interactTutor(): Allows the learner to interact with the smart tutor. For example, the learner may ask for additional explanations of a difficult concept.
  - askQuestions(): Allows the learner to ask questions to the system. For example, the learner may ask for clarification on a programming exercise.
  - realizeExercises(): Allows the learner to complete the proposed exercises. For example, the learner can solve math problems adapted to his or her level.
  - viewProgress(): Allows the learner to see their progress. For example, the learner can see their performance in different skills.
  - customizeEnvironment(): Allows the learner to adjust their learning preferences. For example, the learner can change the pace of learning or the type of exercises he or she prefers.
- Systems Analysis Learning class: The AnalysisLearningSystem class analyzes training data through the analyzeBehavior(), predictPerformance(), identifyDifficult(), recommendResources(), and generate Reports() operations. It provides information to optimize training.
  - analyzeBehavior(): Allows you to analyze learning patterns. For example, it can identify that the learner performs better in the morning or that they spend more time on geometry exercises.
  - predictPerformance(): Allows you to anticipate the learner's future performance. For example, it can predict whether the learner is ready to tackle a complex new concept.
  - identifyDifficulty(): Used to identify blocking points. For example, it can detect that the learner has systematic difficulties with fractions in mathematics.
  - recommendResources(): Allows you to suggest relevant resources. For example, it can suggest additional explainer videos about concepts that are not well understood.
  - generateReports(): Allows you to create detailed analysis reports. For example, it can produce a monthly summary of the learner's progress.
- AgentConvensional class: The Conversational Agent class handles conversational interactions through the analyzeQuestion(), generateAnswer(), maintainContext (), learnInteractions(), and redigerVersExpert() operations. It ensures natural communication with the learner.
  - analyzeQuestion(): Helps to understand the questions being asked. For example, it can identify whether the learner is asking a question about a mathematical concept or asking for clarification.
  - generateAnswer(): Allows you to formulate appropriate answers. For example, it can explain a complex concept with simple analogies.
  - maintainContext(): Helps keep the conversation consistent. For example, it can remember previous questions to provide contextualized answers.
  - learnInteractions(): Allows you to improve your future responses. For example, you can memorize the explanations that worked best.
  - writeToExpert(): Allows you to forward complex questions to the smart tutor. For example, it can redirect a technical question to an expert in the field.
- Learner Profile Class: The LearnerProfile class handles student information through the analyzeProgress(), setObjectives(), updateSkills(), recommendPath(), and generatePortfolio() operations. It maintains a complete view of the learning path.
  - analyzeProgress(): Allows you to evaluate the progress of learning. For example, it can track the improvement of programming skills over a period of time.
  - setGoals(): Allows you to set learning objectives. For example, it can set milestones to reach an advanced level in mathematics.
  - updateSkills(): Allows you to update the mastery level. For example, it can validate the acquisition of a new skill after passing exercises.

   - recommendPath(): Allows you to suggest learning paths. For example, it can suggest an optimal sequence of modules to follow.
   - generatePortfolio(): Allows you to create a portfolio of skills. For example, it can compile achievements and certificates obtained.

- ContentPersonnalize Class: The Content Personalize class handles adapting content through the adaptLevel(), generateExercises(), personalizePresentations(), adjustRhythm(), and enrichContent() operations. It helps ensure that the content matches the needs of each learner.
   - adaptLevel(): Allows you to adjust the difficulty of the content. For example, it can simplify programming exercises if the learner encounters too many difficulties.
   - generateExercises(): Allows you to create custom exercises. For example, it can generate math problems based on the learner's interests.
   - customizePresentations(): Allows you to adapt the format of the content. For example, it can favor visual aids for a learner with a visual learning style.
   - adjustPace(): Allows you to change the pace of learning. For example, it can slow down the progress of a module if the learner needs more time.
   - enrichContent(): Allows you to add additional resources. For example, it can incorporate concrete examples related to the learner's area of interest.

- Resource Manager class: The Resource Manager class organizes educational resources using the indexContent(), searchResources(), optimizeStorage(), update(), and generateStatistics() operations. It manages the system's resource library.
   - indexContent(): Allows you to catalog the available resources. For example, it can categorize educational videos by level and theme.
   - searchResources(): Allows you to quickly find relevant content. For example, it can locate all available exercises on loops in programming.
   - optimizeStorage(): Allows you to manage storage space efficiently. For example, it can archive infrequently used resources.
   - update(): Allows you to update existing resources. For example, it can update exercises with new use cases.
   - generateStatistics(): Used to produce usage analytics. For example, it can identify the most consulted resources.

The Resource Manager class:

- Interacts with the Learner class for participant management

- Contact SmartTutor for group work evaluation

- Uses Resource Manager to store collaborative content

- Integrates with the Learning Analysis System to measure participation

- Collaborative Activity Class: The Collaborative Activity class facilitates collaboration between learners through the createWiki(), editWiki(), participateChat(), createGroup(), shareResources(), and manageVersions() operations. It allows learners to work together and communicate effectively.
   - createWiki(): Allows you to create wiki pages for knowledge sharing. For example, learners can create a wiki page on the key concepts of object-oriented programming.
   - editWiki(): Allows collaborative editing of wiki content. For example, a learner can add practical examples to an existing wiki page on the unified modeling language.
   - participateChat(): Allows you to exchange in real time with other learners. For example, learners can discuss a complex problem or organize their group work.
   - createGroup(): Allows you to form work teams. For example, learners can create a group for a common software development project.
   - shareResources(): Allows you to exchange documents and resources. For example, a learner can share their notes or useful links with their group.
   - manageVersions(): Allows you to track changes made to the wiki. For example, learners can see the history of changes and revert to an earlier version if necessary.

- Group Class: The Group-class handles the academic aspects of a class through the setSchedule(), manageAttendance(), organizeAssessments(), and createWorkGroups() operations. It structures the pedagogical organization of a class group.
   - setSchedule(): Allows you to set the class schedule. For example, they can schedule class sessions and assessments.
   - manageAttendance(): Allows you to manage student attendance. For example, it can track attendance and generate attendance reports.
   - organizeAssessments(): Allows you to organize assessments. For example, they may schedule and coordinate ongoing reviews and assessments.
   - createWorkGroups(): Allows you to create workgroups. For example, he can form balanced teams for collaborative projects.

c) Use case diagram: The use case illustrates the exchanges between a system and its stakeholders (users, other systems, etc.) in terms of usage scenarios and functionalities offered by the system [52]. The identification of user needs and the main functionalities to be developed is often done upstream of the design phase using the use case diagram [53]. In the use case diagram Fig. 2, the actors are symbolized by external blocks while the use cases are indicated by ellipses. The arrows symbolize the links between the actors and the usage scenarios [54].

- Diagram interpretation:

Learner: The learner is at the center of the system and interacts with several functionalities. He can:

- Interact with the tutor: This suggests a built-in tutoring system, likely AI-driven, that offers personalized assistance.
- Check your progress: The learner can track their performance and progress in the course.
- Customize the environment: The system offers customization options to tailor the learning experience to the learner's preferences.
- Create/Edit a wiki: A collaborative feature for learners to contribute to a common knowledge base.
- Participate in a chat: Facilitates communication and interaction between learners.
- Create a group: Allows learners to form workgroups for collaborative projects or studies.

Teacher: The teacher supervises the learning process and has access to the following functions:

- Set up the AI tutor: The teacher can adjust the settings of the AI tutor to optimize its effectiveness.
- Track learning: The teacher can track learners' progress and identify difficulties.
- Analyze the results: The teacher can examine the performance of the learners and evaluate the effectiveness of the teaching.
- Intervene if necessary: The teacher can intervene to provide additional support or personalized advice.

System Analysis Learning: This component represents the analytical backend of the system. It does the following:

- Analyze behavior: The system tracks and analyzes learners' actions in the learning environment.
- Predict performance: Based on the data collected, the system can predict the future performance of learners.
- Identify difficulties: The system can detect areas where learners are struggling.
- Recommend resources: Based on the needs identified, the system may recommend additional learning resources.

Chatbot: This actor represents a chatbot or virtual assistant. Its actions include:

- Analyze the question: Understand the questions asked by the learners.
- Generate Response: Provide relevant answers to questions.
- Maintain context: Maintain the context of the conversation for more natural interactions.

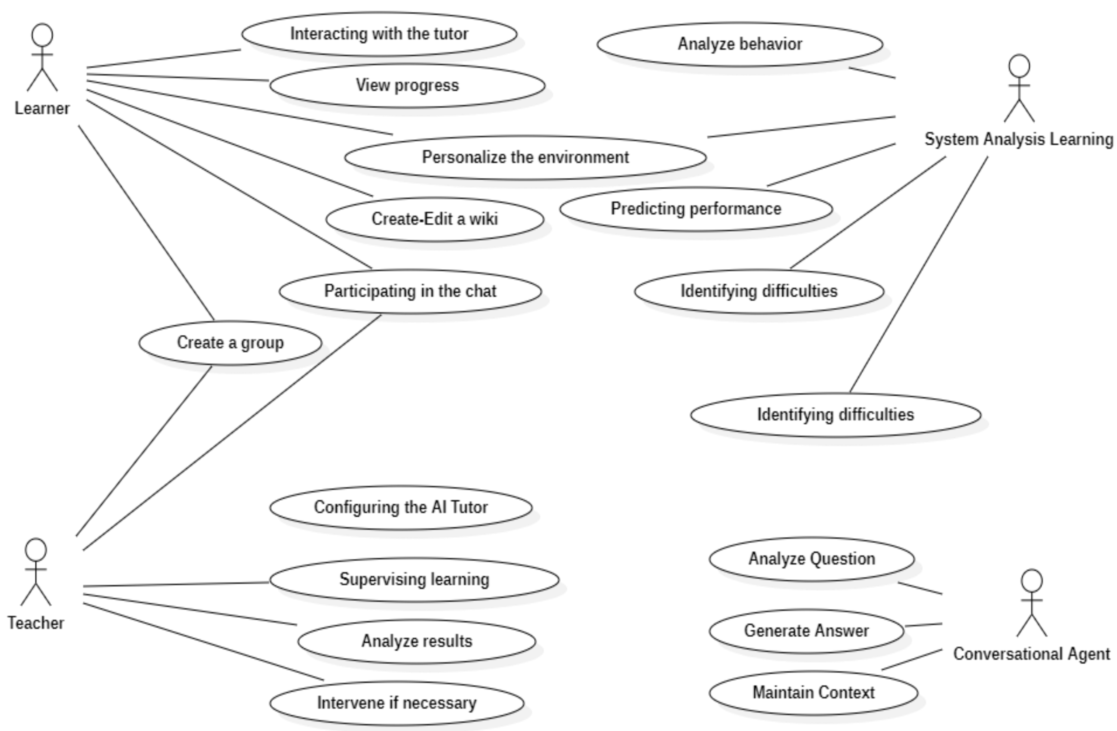

Fig. 2. Use case diagram and interactions between the different actors.

2) Dynamic part: UML dynamic modeling offers a complementary vision of the behavior of a system [55]. While sequence diagrams emphasize the chronological order of interactions between different objects, activity diagrams present a more holistic view of processes, breaking down actions into sequences, parallels, or conditions. They thus provide a better understanding of how the system works, identify potential problems and facilitate communication between the different actors of a project [56]. There are two types of diagrams: the sequence diagram and the activity diagram.

a) The Activity diagram: A UML activity diagram is a graphical representation that visualizes the control flow of a process, procedure, or algorithm [57]. It makes it possible to model in detail the different stages of a process [58]. The decisions to be made at each stage, the loops and the synchronizations.

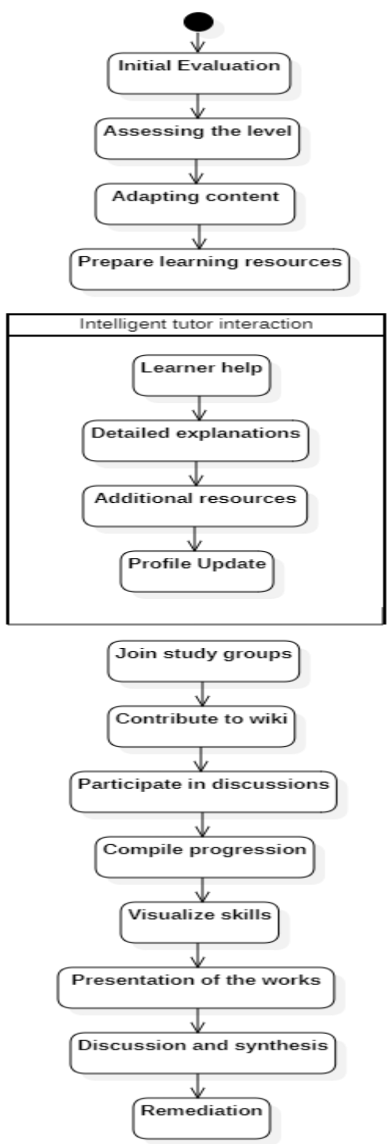

Fig. 3. Activity diagram describing the workflow within the learning system.

- Interpreting the intelligent learning system presented in this activity diagram (Fig. 3), illustrates a comprehensive and sophisticated learning process. It all starts with a crucial initial assessment phase that precisely identifies the learner's level, followed by an in-depth assessment to tailor the learning content to their specific needs. Once this foundation is established, the system prepares customized learning resources. Interaction with the smart tutor is at the heart of the system, offering targeted assistance, detailed explanations and additional resources, while keeping the learner's profile up to date. The system strongly encourages the collaborative dimension of learning by inviting learners to join study groups, contribute to a collective wiki, and actively participate in discussions. Progress is monitored rigorously, with a regular compilation of progress, a clear visualization of the skills acquired and presentations of the work carried out. The process concludes with discussion and synthesis sessions, allowing to consolidate learnings, with the possibility of implementing corrective actions if necessary. This methodical approach combines personalized learning, intelligent interaction, and social collaboration, creating a comprehensive and effective learning environment that continually adapts to the learner's needs.

b) A sequence diagram: According to Damy, the UML sequence diagram is a visual representation that illustrates the interactions between different objects in a system over time [59]. It makes it possible to model in detail the exchange of messages between these objects, following a precise chronology [60]. Each object is represented by a vertical lifeline, and the messages exchanged are represented by horizontal arrows. These diagrams are particularly useful for understanding the dynamics of a system, identifying synchronization points, and analyzing usage scenarios.

- Interpreting: The sequence diagram (Fig. 4), illustrates the dynamic and chronological interaction between the different actors in the learning system: the teacher, the learner, the group of learners, the class group and the system itself. The sequence highlights the message exchanges and communication flows between these entities. The learner initiates the process by registering and defining their profile, followed by interactions with the learning content through learning activities and assessments. The system plays a pivotal role in managing resources, processing responses, and providing automated feedback. The collaborative dimension is represented by the interactions between the learner and his or her group, as well as with the whole class, especially during discussions and shared activities. The teacher intervenes at key moments for validation and final evaluation. This sequential representation places particular emphasis on the temporal aspect and the sequencing of interactions, showing how each action triggers a specific reaction or response from other actors in the system, thus creating a coherent and structured learning flow.

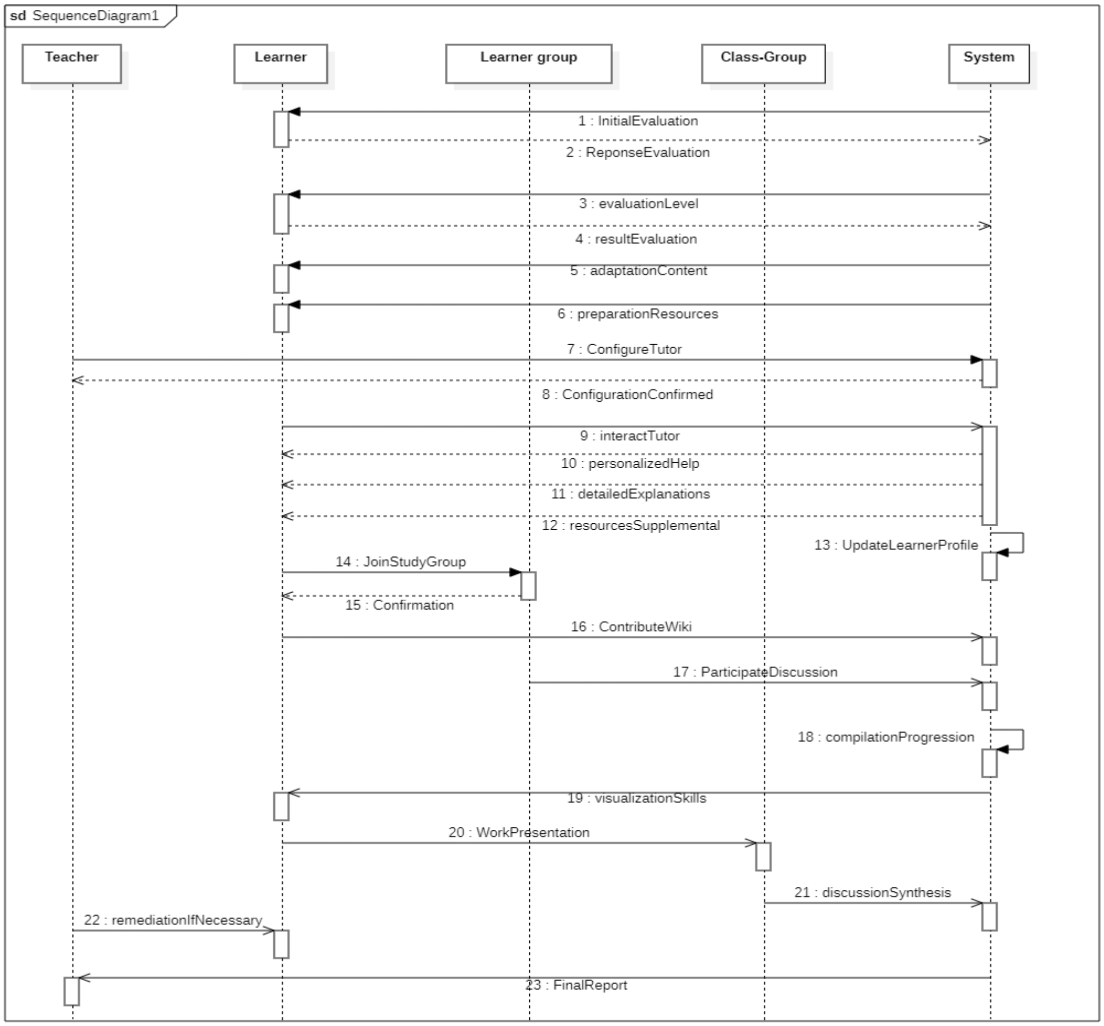

Fig. 4. Sequence diagram and interactions between system objects.

From the foregoing, the UML modelling of an integrated learning system combining adaptability, collaboration and artificial intelligence is a significant advance in the field of digital educational environments. This model demonstrates how it is possible to harmonize the personalization of learning paths with the activation of social interactions, two aspects frequently examined separately in current systems. The clear and modular structure offered by UML not only facilitates design, but also improves communication between the different actors, ranging from creators to teachers, as well as technical developers. This strategy better meets the challenging requirements of personalized, collaborative learning at scale.

The incorporation of artificial intelligence as a central element of the system represents a real lever for optimizing educational pathways and perfecting support. By examining learners' behavioral and interactive data in real time, AI can refine content, organize relevant collaborations, and provide tailored feedback. This capacity for dynamic adjustment far exceeds that of traditional adaptive systems anchored in rigid rules, offering a more responsive and resilient framework. In addition, the automation of repetitive tasks by artificial intelligence relieves teachers of certain administrative obligations, allowing them to focus more on supporting and designing innovative pedagogies. However, while promising, this modeling also presents challenges: the complexity of managing various adaptive and collaborative modules simultaneously requires a robust and scalable technical architecture that can adjust to technological and pedagogical advances. The modular approach adopted offers essential flexibility, opening up prospects for the subsequent integration of new functionalities, such as learning through gamification, predictive analysis of the risk of dropping out or the use of more advanced intelligent conversational agents. These perspectives accentuate the potential of this model to become a sustainable part of digital pedagogical practices, by meeting the growing needs for personalization and remote collaboration.

Finally, this work enriches the reflection on the creation of more intelligent and humane e-learning systems, by finely articulating complementary dimensions in a coherent structure. It encourages further research and experimentation to better understand and exploit the synergies between adaptability, collaboration and artificial intelligence for the benefit of more inclusive and effective education.

IV. Challenges and Prospects

Developing an integrative online learning system based on adaptivity, collaboration, and AI poses several challenges:

- technical interoperability,

- balancing algorithmic personalization and mediation.

For the future, the integration of immersive technologies and generative AI offers promising prospects, provided that collaboration between researchers, designers, and educators is strengthened to ensure inclusive, sustainable, and pedagogically relevant learning environments.

V. Conclusion

UML modeling of an integrative learning system, combining adaptability, collaboration and artificial intelligence, represents a major step towards creating more efficient, personalized and dynamic online educational environments. This modular framework facilitates the clear and logical organization of complex interactions between learners, content, and smart technologies, while providing essential flexibility to meet diverse user needs. The combination of personalizing journeys and encouraging social interactions, supported by sophisticated artificial intelligence mechanisms, not only improves the quality of individual learning, but also underlines the collective and collaborative dimension, which is crucial for motivation and success.

However, this innovative approach also presents considerable challenges. On the technical side, the complexity of adaptive algorithms and interaction analysis modules requires rigorous design and exhaustive validation to ensure the reliability, robustness and scalability of the system. From a pedagogical point of view, it is essential to find a balance between automation and human intervention, in order to maintain an educational mediation attentive to the emotional and cognitive needs of learners. In addition, ethical issues related to the use of personal data, the transparency of algorithmic processes and the prevention of bias must be addressed with sustained attention, thus ensuring a responsible and fair use of artificial intelligence in the field of education.

Despite these challenges, the proposed model represents a promising framework that can be adapted and expanded to different educational contexts and disciplines. It paves the way for future research to experiment and refine these architectures in real-world environments, while assessing their impact on learning, collaboration, and motivation. This work thus serves to enrich the reflection on the digital transformation of education, by putting people at the center of technological innovations and by promoting a more inclusive, agile and intelligent pedagogy.

The future of online learning systems rests on this ability to effectively combine advanced technologies and a student-centered approach to meet the complex and evolving challenges of distance and hybrid learning in an increasingly connected and demanding world.

References

[1] Dridi, C. E., Benzadri, Z. and Belala, F. (2020, June). Systems engineering system: meta-modeling perspective. In 2020, IEEE 15th International Conference of System of Systems Engineering (SoSE) (pp. 000135-000144). IEEE.

[2] VanLehn, K. (2011). The relative effectiveness of human tutoring, intelligent tutoring systems, and other tutoring systems. Educational Psychologist, 46(4), 197–221.

[3] Brahmi, M., Louati, L., & Ghédira, K. (2023). Adaptive and collaborative e-learning systems: A comprehensive review of personalization, social interaction and AI integration. International Journal of Educational Technology in Higher Education, 20(1), 1–20. https://doi.org/10.1186/s41239-023-00423-z

[4] Aberšek, B. (2024). Artificial intelligence and the transformation of education: A cognitive–constructivist approach. Educational Technology Research and Development, 72(1), 115–132.

[5] Possati, L. (2021). The Algorithmic Unconscious: How Psychoanalysis Helps in Understanding AI. AI & Society, 36(4), 1125–1135. https://doi.org/10.1007/s00146-021-01161-8

[6] Gao, Y., Liu, Z., Si, Y., Meng, Y., Xiong, C., & Lin, Y. (2023). GEO: Generative Engine Optimization. arXiv preprint arXiv:2311.09735. https://arxiv.org/abs/2311.09735

[7] Niang, M., Mbaye, S., & Diop, M. (2022). Adaptive and Collaborative E-Learning Systems: Integrating Artificial Intelligence for Enhanced Learning Experiences. Journal of Educational Technology and Society, 25(3), 45–60.

[8] Delès, M. (2021). Digitalisation pédagogique et défis de l'enseignement à distance : Cas de l'École Supérieure de Technologie d'Agadir. Revista Multidisciplinar, 4(1), 27–48. https://doi.org/10.23882/rmd.22083

[9] Maha, K., Omar, E., Mohamed, E., & Mohamed, K. (2021). Design of educational scenarios of activities in a learning situation for online teaching. GSC Advanced Engineering and Technology, 1(1), 049-064.

[10] Basque, J., Paquette, G., & Henri, F. (2022). Digital Learning Environments and Pedagogical Engineering: What are we Talking About?. Learning and Teaching on the Web: What Pedagogical Engineering?, 23.

[11] Vygotsky, L. S. (1978). Mind in Society: The Development of Higher Psychological Processes. Cambridge, MA: Harvard University Press.

[12] Johnson, D. W., & Johnson, R. T. (2009). An Educational Psychology Success Story: Social Interdependence Theory and Cooperative Learning. Educational Researcher, 38(5), 365–379.

[13] Dillenbourg, P. (1999). What do you mean by collaborative learning? In P. Dillenbourg (Ed.), Collaborative learning: Cognitive and computational approaches (pp. 1–19). Oxford: Elsevier.

[14] Kirschner, P. A., Sweller, J., & Clark, R. E. (2009). Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. Educational Psychologist, 41(2), 75-86.

[15] Scardamalia, M., & Bereiter, C. (2006). Knowledge building: Theory, pedagogy, and technology. In K. Sawyer (Ed.), The Cambridge handbook of the learning sciences (pp. 97–115). Cambridge University Press.

[16] Strijbos, J. W., & De Laat, M. (2010). Developing the role concept for computer-supported collaborative learning: An explorative synthesis. Computers in Human Behavior, 26(4), 495–505.

[17] Slavin, R. E. (1980). Cooperative learning. Review of Educational Research, 50(2), 315-342

[18] Shute, V. J., & Zapata-Rivera, D. (2012). Adaptive educational systems. In D. Jonassen & S. Land (Eds.), Theoretical foundations of learning environments (2nd ed., pp. 277–294). Routledge.

[19] Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. Engineering education, 78(7), 674–681.

[20] Pashler, H., McDaniel, M., Rohrer, D., & Bjork, R. (2008). Learning styles: Concepts and evidence. Psychological Science in the Public Interest, 9(3), 105–119.

[21] Brusilovsky, P., & Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In P.

[22] Conati, C., & Kardan, S. (2013). Student modeling: Supporting personalized instruction, from problem-solving to exploratory open-ended activities. AI Magazine, 34(3), 13–26.

[23] Bloom, B. S. (1968). Learning for mastery. Instructional objectives, 1(2), 5-8.

[24] Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: A model and seven principles of good feedback practice. Studies in Higher Education, 31(2), 199–218.

[25] Hattie, J., & Timperley, H. (2007). The power of feedback. Review of Educational Research, 77(1), 81-112.

[26] Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent Web-based educational systems. International Journal of Artificial Intelligence in Education, 13(2-4), 1-38.

[27] Baker, R. S., & Siemens, G. (2014). Educational Data Mining and Learning Analytics. In R. K. Sawyer (ed.), The Cambridge Handbook of the Learning Sciences (2nd ed., pp. 253-274). Cambridge University Press.

[28] Chen, C. M., Huang, T. C., & Lee, P. Y. (2020). AI-enabled personalized learning: A review of the literature. Journal of Educational Computing Research, 58(2), 223-247.

[29] Rosé, C. P., Wang, Y. C., Cui, Y., Arguello, J., Stegmann, K., Weinberger, A., & Fischer, F. (2008). Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning. International Journal of Computer-Supported Collaborative Learning, 3(3), 237–271.

[30] Baker, R. S. J. d., & Inventado, P. S. (2014). Educational data mining and learning analytics. In J. A. Larusson & B. White (Eds.), Learning analytics: From research to practice (pp. 61–75). Springer.

[31] Roll, I., & Wylie, R. (2016). Evolution and revolution in artificial intelligence in education. International Journal of Artificial Intelligence in Education, 26(2), 582-599.

[32] Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). Intelligence unleashed: An argument for AI in education. Pearson.

[33] Binns, R. (2018). Fairness in machine learning: Lessons from political philosophy. Proceedings of the 2018 Conference on Fairness, Accountability and Transparency, 149–159.

[34] Holmes, W., Bialik, M., & Fadel, C. (2019). Artificial Intelligence in Education: Promises and Implications for Teaching and Learning. Center for Curriculum Redesign.

[35] Ba, D. (2023). Design and implementation of a girls' monitoring application at the Maison De l'Education.

[36] Bodiang, A. (2018). Design of an application for the automation of land use planning for the Société Civile Immobilière "LE TOIT DU SUD".

[37] Diatta, A. A. I. (2019). Design and development of a charging management application for UASZ staff.

[38] Batoure, A. B. (2020). Use of model-driven engineering for modeling, implementing, and powering decision support systems. International Journal of Data Engineering (IJDE), 8(1), 1-12.

[39] Rosenberger, M. (2022). UML 2.x: From fundamentals to advanced modeling techniques. Springer.

[40] Ouelhadj, S., Champin, P. A., Jean-Daubias, S., & Gaillard, J. (2023, January). Method of semantic data enrichment using annotated UML. In EGC 2023-Knowledge Extraction and Management (pp. 635-636).

[41] Germain, L., Dufresne, J., & Trépanier, A. (2024). Visual modeling and collaboration in software teams: The enduring role of UML. Software Practice and Experience, 54(1), 45–62.

[42] Marmoul, T. E., Belaïd, R., & Hossain, A. (2024). Extending UML for adaptive system design: A survey and framework proposal. International Journal of Advanced Computer Science, 35(1), 87–103.

[43] Conati, C., Jaques, N., & Muldner, K. (2021). AI in education: Adaptive learning and affective computing. In R. Nkambou, R. Azevedo, & J. Vassileva (Eds.), Advances in Intelligent Tutoring Systems (pp. 55–72). Springer.

[44] Millán, E., & Chen, Y. (2023). Intelligent adaptive learning: Towards granular personalization with AI techniques. Computers & Education: Artificial Intelligence, 4, 100131.

[45] Kreijns, K., Kirschner, P. A., Vermeulen, M., Buuren, H., & Acker, F. (2022). The myth of the digital native and the multi-tasking learner: The reality of collaborative learning online. Educational Technology Research and Development, 70(1), 23–42.

[46] Dillenbourg, P. (2021). Orchestrating Collaborative Learning: Contemporary Perspectives. Springer.

[47] Holmes, W., Bialik, M., & Fadel, C. (2022). Artificial Intelligence in Education: Promises and Implications for Teaching and Learning (2nd ed.). Center for Curriculum Redesign

[48] Roschelle, J., & Teasley, S. D. (1995). The construction of shared knowledge in collaborative problem solving. In C. E. O'Malley (Ed.), Computer supported collaborative learning (pp. 69–97). Berlin: Springer.

[49] Dennis, A., Wixom, B. H., & Roth, R. M. (2020). Systems Analysis and Design (7th ed.). Wiley.

[50] Müller, C., & Reichenbach, K. (2014). Adaptive Learning Systems: Principles and Implementation. Journal of Educational Technology & Society, 17(3), 45–59.

[51] Chehida, S., Baouya, A. and Bensalem, S. (2021, October). Component-based approach combining UML and BIP for rigorous system design. In International Conference on the Formal Aspects of Component Software (pp. 27-43). Cham: Springer International Publishing.

[52] Latte, M. R., & Javalagi, C. M. (2019, December). Theoretical and statistical analysis of inventory and warehouse management in supply chain management – a case study on small industries. In International Conference on Advanced Production and Industrial Engineering (pp. 597-608). Singapore: Springer Nature Singapore.

[53] AL-Msie'deen, R. F., Blasi, A. H., Salman, H. E., Alja'afreh, S. S., Abadleh, A., Alsuwaiket, M. A., ... and Al-Showarah, S. A. (2022). Detect commonalities and variability in use case diagram variants. arXiv arXiv preprint: 2203.00312.

[54] Ochodek, M., Kopczyńska, S., & Staron, M. (2020). Deep learning model for end-to-end approximation of COSMIC functional size based on use case names. Information and Software Technology, 123, 106310.

[55] Babaalla, Z., Jakimi, A., Oualla, M., Saadane, R., & Chehri, A. (2024, April). Towards a self-extracting UML class diagram from the textual system specification. In the proceedings of the 7th International Conference on Networks, Intelligent Systems and Security (pp. 1-5).

[56] Cormi, C., Abou Amsha, K., Tixier, M., and Lewkowicz, M. (2024, May). Modeling for Analysis and Design in Regulated Artifact Ecologies (MADRAE): A Case Analysis of Cooperative Telemedicine Practices. In 42nd INFormatic Congress of Organizations and Information and Decision Systems (INFORSID 2024) (pp. 11-12).

[57] BELKACEM, B., AND HICHAM, B. (2023). The automatic transformation of activity diagrams into Petri nets (PhD thesis).

[58] Diamanka, M. (2024). Access control and appointment scheduling system for companies outside the logistics distribution platform of the Autonomous Port of Dakar.

[59] Damy, S. (2024). Lectures and tutorials in UML modeling (introduction) (Doctoral thesis, Centre de téléenseignement universitaire de Franche-Comté).

[60] Bikengela, T. A. (2024). Modeling and Design of Adaptive Software Systems: Approaches and Tools. Éditions Universitaires, Paris.