Collaborative Tutoring Architecture: A Generic Case Based Reasoning Multi-Agent

M. Ennaji, (Labsiv, University of Agadir, Morocco), H. Boukachour (LITIS, University of Le Havre, France), M. Machkour (Labsiv, University of Agadir, Morocco)

***Abstract*—This article is a response to the problem of learner desertion encountered by e-learning platforms. We propose to equip the Learning Management System with a generic intelligent tutoring module. This module is based on the combination of Case Based Reasoning (CBR) and Multi-Agent Systems (MAS). The main advantage of this generic module is to offer the learner individualized follow-up and to prevent dropping out of school. Personalized monitoring is carried out by combining machine tutoring and human tutoring. We describe how the combination of CBR and MAS allows the adaptation of the learning process according to the profile of the student.**

Index Terms—**Intelligent Tutoring, affective learning, dynamic Case Based Reasoning, Multi-Agent System, generic model.**

I. Introduction

The number of students in universities has only increased. This massification does not allow for personalized monitoring of students. This generates a very high rate of students in greats difficulties resulting in a high dropout rate. Undergraduate studies are characterized by a higher evaporation rate in the first year compared to other possible courses in higher education.

To address these persistent inequalities, these numerous failures or dropouts, changes in the transmission of knowledge and the social-economic challenges of higher education, new learning approaches are emerging and put learners at the center of the learning system, both inside and outside the classrooms, both face-to-face and remote (e-learning).

The impossibility of meeting physically, dispersion and distance from training centers are elements that lead to an acceleration of distance learning practices. With the advent of open and massive online courses, online training must provide for the presence of a tutor alongside the learners. Otherwise, such training is reduced to simple online courses, which has a fairly reduced utility and does not allow students to overcome their possible difficulties.

A study was conducted at the Community College Research Center (CCRC) of Columbia University, involving 40,000 students from different backgrounds who took thousands of online courses in Washington State over a period of 5 years. The researchers measured the attendance and academic performance of these students, comparing them to their face-to-face counterparts. The study found that distance learning had higher drop-out rates than face-to-face training. Only 91% of online students complete their course compared to 94.5% in face-to-face classes [58].

Digital networks allow learners to access learning software, whether specific or offered as web applications. Web-based learning enables more students to have access to the distance-learning environment, and provides students and teachers with flexibility. However, using this learning means exposes a few problems. Among others, teachers accustomed to traditional teaching methods often find it difficult to put their courses online, and some students find themselves overloaded with too much information.

The desertion rate is higher when compared to that of a face-to-face course. How can we minimize this risk in the process of acquiring knowledge?

A major challenge in higher education is to improve both the productivity of teaching and the quality of learning for a large and diverse population of students facing real world constraints such as limited financial resources and a sufficient number of qualified instructors. The literature in education suggests that students who are actively engaged in the learning process are more likely to be successful.

Intelligent Tutoring Systems (ITS) possess a vast knowledge on certain matter, and their role is to transmit this knowledge to the students by means of an interactive individualized process, trying to emulate the form in which a tutor or human teacher would guide the student in his/her learning process.

One of the main issues of Intelligent Tutorial Systems (ITS) consists in adapting to the needs of the student who is interacting at all times. One way to adapt the user is through so-called instructional strategies, which specify how to sequence content, what type of feedback should be given during teaching, when and how tutorial content should be shown or explained.

In this article we propose to increase the functionalities of LMS (Learning Management System) with an intelligent tutoring

Mourad Ennaji, Computer Systems and Vision Laboratory (Labsiv) University Ibn Zohr, BP 8106 - Cité Dakhla 80000 Agadir, Morocco (m.ennaji@uiz.ac.ma)

Hadhoum Boukachour, Computer Science, Information Processing and Systems Laboratory (LITIS) University of Le Havre 25 Rue Philippe Lebon, 76086 Le Havre Cedex, France (Hadhoum.Boukachour@univ-lehavre.fr)

Mustapha Machkour, Computer Systems and Vision Laboratory (Labsiv) University Ibn Zohr BP 8106 - Cité Dakhla 80000 Agadir, Morocco (m.machkour@uiz.ac.ma)

module. This involves rethinking the learning process in order to integrate into LMSs a module offering joint Human/Machine tutoring. The tutoring proposed must include machine tutor but also teacher and groups of learners following the same course.

We present an intelligent system, based on Multi-Agent Case-Based Reasoning, to support student-centered, self-paced and highly interactive learning. The system provides a rich set of online content, maximizes the interactivity between the intelligent learning system and students, and customizes the learning process to suit the needs of each student. In the system, profiles related to the student's learning, such as learning styles and basic knowledge, are used to select, organize and present the learning material to each student and to support the active learning.

The strength of our approach lies in the integration of the human dimension in the computer system. An active collaboration of the Human/Machine relationship allows the different actors to interact intelligently in the learning process. In this way, the learner becomes an active actor who participates in the realization of his learning and the teacher can deploy his skills to overcome the limitations of Machine tutoring.

According to the learner's profile, the ITS performs a diagnosis and an adaptation of the tutoring process. Thus, a research objective is to develop an adaptive intelligent tutoring module. Adaptability can be achieved in several ways. The originality of our approach is to offer a learner support strategy based on an articulation / collaboration of tutors (machine, peers and teacher).

We present a collaborative and generic tutoring module. The paper is organized as follows: after related works, we present the research issue, we then detail our approach, then the architecture before concluding.

II. Related Works

In recent years, education has been characterized by the promotion of independent study. This form of learning is supported by technological systems, such as the Learning Management System (LMS) [32] [35] and the Intelligent Tutoring System (ITS) [59].

Tutoring is a form of teaching that has two main features compared to classroom teaching. The first characteristic concerns the tutor/student ratio, usually 1:1 (or 1/2, 1:3). The tutor's attention is therefore devoted to one student at a time. The second characteristic concerns the tutor's guiding role in the teaching. Tutoring plays an inverse role to that of classroom teaching. In the classroom, the teacher asks each student to adapt to a common course for the whole class, while the tutor tries to adapt his or her intervention to the needs of one student [13].

Following the various publications on tutoring and its effectiveness [12] [18] [60], the AIDE (Artificial Intelligence in Education) research community has used the notion of tutoring to develop Intelligent Tutoring Systems. ITSs are computerized learning environments that aim to imitate and simulate the behavior of a human tutor in his or her capacities as an expert pedagogue and expert in the field. As with teaching, the two main functions of tutoring are:(i) to elicit learning and (ii) to evaluate it. In ITS, these two functions are dealt with separately or jointly.

ITS have being studied by educational and computer science researchers since the 80s [3]. ITS have been built for various scientific domains such as computer programming [25] [57], engineering [14] [36], mathematics [55], physics [43], chemistry [9], medicine [53], etc.

For Hafner [27], ITS is an intelligent teaching software that follows the evolution of students' work and offers them personalized feedback. By analyzing the work of a given learner, the software can suggest ways to guide that learner according to his or her strengths and weaknesses.

According to Woolf [63] and Nkambou and al [45], ITS consists of four main parts (Fig. 1):

1. domain model: refers to the expert's knowledge of the domain and the object being taught;
2. learner model: represents the learner's skills and actions;
3. tutor model: makes choices of pedagogical assistance based on the two previous models;
4. interface: allows interaction between the learner and the system.

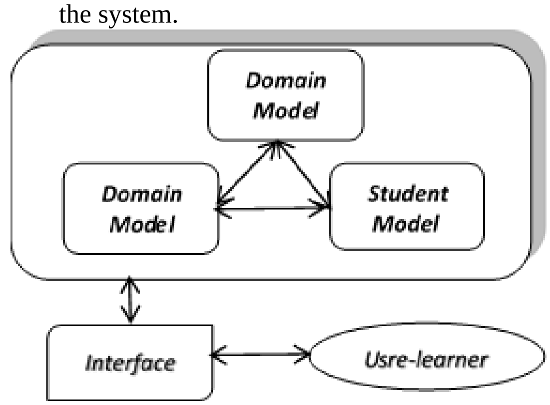

Fig. 1. ITS Components [45]

The real-time adaptation of the teaching situation to the learner is one of the major objectives of ITS. To achieve this, ITS is based jointly on the learner model and the tutor model. Classically, the "learner model" refers to what the learner knows, what he has done, his learning strategies etc. The information represented by this model can be the learner's skills, meta-cognition and emotional characteristics. This information is usually an inference that the system makes about the learner. These inductions are constructed by observing the interactions that the learner has with the system and by measures of learner performance such as the time taken to complete a task and the errors observed. The learner's model does not make any decisions, it only provides information to the tutor's model so that the tutor can adapt his interventions to the learner.

Based on knowledge of the domain and learner model, the tutor model monitors the interactions between the system and the learner on an ongoing basis to ensure that the tutor's strategies are adapted to the learner. Tutor behavior needs to be executed in real time and the main challenge of the tutor model is to identify when and how to intervene to help the learner [49].

Adapting the learning environment to meet the specific needs of each learner is the expected objective of ITS. As a result, the use of this pedagogical strategy is applicable to a wide range of areas. In this sense, there are several ITS but they are specific to a single subject (geometry, health, etc.) [51], [56].

The development of tutoring in ITS, whether it involves services, components or functions integrated into another component, expresses the choices that are made in terms of paradigms of cognition.

There are specialized software environments for the development of ITS called authoring systems. These author systems are also associated with a paradigm. In this sense, sharing and reuse are limited to systems in the same category. In addition to authoring systems that aim to develop the whole system, some tools specialize in one component [54]. Some simplified, high-level, paradigm-specific authoring tools have been developed to increase accessibility and reduce development costs [49].

It is important for education systems to ensure that learners learn, but above all to look after their emotional and affective state like a human tutor [4]. Emotionality in learning is defined as the response given by a tutor, with a behavior or dialogue, which produces a benefit in the students [12]. The sine qua none condition for integrating affectivity into the process is to detect the emotional states of learners in an active way during tutoring [24] [30].

Based on distance learning experiences during the Covid crisis 19 and the work of researchers [7] [39], we are now sure that the affectivity and human dimension would considerably improve learner motivation and the learning process. For this reason, it is necessary to include certain affective factors and to propose a psychological model of the student.

Several works have attempted to arouse emotion in the student. Bertola and al [11] presented a method based on an ontological approach. In addition to ontology, [31] used a formal representation of the phenomenon, mathematical assessment and empirical assessment with undergraduate students. Similarly, Balakrishnan and al [8] developed an approach to modeling a student's affective state. On the other hand, Arguedas and al. [5] proposed a model that includes different types of emotions, moods and behaviors of students in online learning environments.

We believe that these approaches based on the modeling of the affective state by machine do not replace the primary role and place of the human guardian. From this premise, the integration of the human (teacher or peers) is an important factor in overcoming the limitations of machine tutoring.

III. Proposed approach

A. Generic ITS

We propose such a system in addition to conventional education. It is about providing learners with an additional means for their learning and teachers with a device to better follow the learners. For the teacher, the ITS will offer him the means to devote himself more to learners who present difficulties and who require support that machine tutoring or peers cannot provide. The question of learner autonomy in hybrid situations will be one of the key issues in didactic and computer modelling.

The genericity and articulation of tutoring is a solution to resolve some of the limitations of current ITS. In this paper, we propose an original idea to develop generic ITS. Our approach is composed of three modules (Fig. 2):

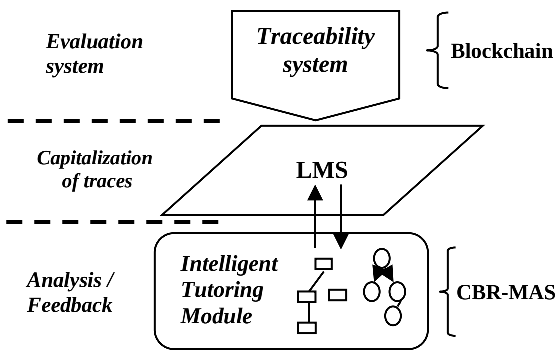

Fig. 2. Generic ITS model

1. Learning module: use of an existing learning platform to retrieve the learner's traces.
2. Intelligent tutoring module: responsible for equipping existing LMSs with a module capable of offering personalized, individualized tutoring that is independent of the learning subject. In addition, this module provides a link between machine tutoring and human tutoring.
3. Certification module: The proposed ITS will address the following two functions: initiating learning and evaluating it. The assessment will result in certification of the skills acquired. We will set up a certification mechanism so that learners can more easily value their skills with companies and therefore achieve better employability. For its implementation, we will rely on Blockchain technology which has properties of tampering and traceability of records made in a decentralized register. In addition, thanks to the use of Smart Contracts, it will be possible to automate the attribution of certificates following the evaluations carried out within the ITS. A Blockchain interrogation tool will allow companies to verify the authenticity of certificates presented by job applicants. This skill certification mechanism will be completely independent of the learning object.

However, this article focuses on the intelligent tutoring module.

B. Articulation between machine/human tutoring

Tutors are more than common transmitters of information; they are responsible for creating a suitable environment for learning [50]. Tutors are responsible for guiding learning and teaching new knowledge to students. It is important that these tutors have emotional skills when teaching students.

ITS are traditionally built around an artificial tutor. It has expertise in a particular area of knowledge and applies a teaching strategy to interact with a learner to help him or her solve a given problem. This principle of the learner-machine couple operating independently can be satisfactory until the system reaches its limits; the presence of a human or peer teacher then becomes indispensable. Peers who have already integrated advanced notions and can therefore help students with difficulties. Their approach could be beneficial because it is different from that of the teacher in charge of the course.

The problem is to integrate the analysis of the pedagogical needs of the learners and the taking into account of the relations between peers and teachers within an ITS. The role of the ITS is to offer the learner a personalized follow-up adapted to their needs and skills.

To respond to this problem, it requires working on an articulation between "machine" tutoring and "human tutoring" (teachers and peers). "Human" tutoring can indeed come both from teachers, but also from other learners who have already integrated advanced concepts and could therefore come to the aid of learners in difficulties. Their approach could be beneficial because it is different from that of the teacher in charge of the course. Ultimately, it is about designing a system essential to the teacher's activity and placing learners at the heart of active learning, by offering them tools for investigative learning methods and collaborative workspaces.

C. ITS Multisubject

Existing ITS are designed and developed for a specific learning subject. This solution is time consuming and tedious. An alternative, reliable and sustainable solution to support students and meet their needs would be to develop an ITS to provide personalized tutoring considering all the learner's difficulties. This involves adding together the different knowledge about the learner (difficulties, skills, prerequisites, etc.) from all the subjects in order to offer him feedback custom-made. The learner will thus have a single ITS for his learning journey, several learning objects, with tutoring based on an articulation between machine, peers and teachers.

The idea is to add an intelligent tutoring module, multi-tutor and multi-subject, to any learning platform. Thus, with this module, an LMS will have an additional functionality, that of offering the learner personalized follow-up for the entire course. Using a multi-tutor, multi-subject intelligent tutoring module could significantly reduce the failure or dropout rate of students.

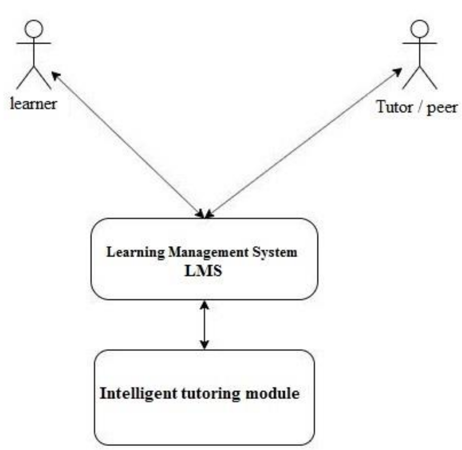

Fig. 3. Augmented Learning Management System

We propose to implement a generic intelligent tutoring module whose aim is to offer machine and human tutoring. The aim is to increase the LMSs of a module which combines human and machine tutoring (Fig. 3).

D. Description of the Intelligent Tutoring module

To address the lack of ITS capable of offering the learner a tutoring space for all the subjects that are part of his learning path and multiple forms of tutoring (machine, peers and teachers), the failure and dropout rate is higher compared to conventional classroom-based course. It is necessary to offer learners a single ITS for their learning pathways and thus avoid the use of various ITS. An ITS by subject would not take into account the skills and difficulties of the learner at the course level but only at the subject level. Therefore, a generic ITS independent of the learning subject would be needed and therefore the domain model used should be too.

We propose an approach for the development of an intelligent tutoring module based on the interaction of two parts (Fig. 4):

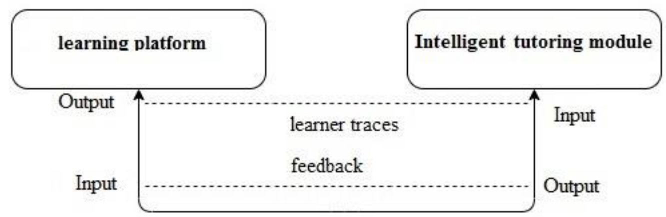

Fig. 4. Interaction LMS and Intelligent Tutoring Module.

1. Learning: use of an existing LMS to retrieve the learner's traces.
2. Intelligent tutoring: responsible for providing the LMS with a module capable of offering personalized tutoring independent of the learning subject. It provides a link between « machine » tutoring and « human » tutoring.

Our goal is to increase the LMS of a module that combines several types of tutoring (machine, peers and teachers). This module is divided into two parts: the diagnosis of the learner's knowledge (for example, the detection of the causes of errors) and the choice of remedial strategies [13]. The intelligent tutoring module will make it possible to detect students' difficulties and offer them appropriate follow-up. The "machine" tutor will help the student with basic learning that does not require human intervention. This module is based on the analysis of the educational needs of learners and also taking into account relations between peers. Thus, the teacher's role is to devote full attention more to learners who require support that machine tutoring or peers cannot provide them.

Designing and Developing an ITS is a difficult task. There are three categories of methods to achieve the tutoring function:

1. metacognition-based methods.
2. methods based on artificial intelligence and trace analysis.
3. methods based on cognitive architectures.

Our model uses IA and trace analysis techniques. The combination of artificial intelligence (AI) and education has led to the development of several intelligent educational software for different fields. Machine Learning techniques provide a variety of methodologies and theories on reasoning, inference, and learning. Machine learning-based ITS can adapt the course to the knowledge, experience, strengths and weaknesses of the learner. ITS are complex to build and maintain and face the difficulty of acquiring knowledge.

We do not seek to model the mechanisms of human learning, but we aim to identify, according to the actions of the learner, the new knowledge to be brought to him. Bayesian networks is one of the techniques which makes it possible to carry out this type of tutoring [41] [48]. This involves detecting, during the learning activity, behaviors that may present educational risks, and identifying them in relation to existing cases. Then determine if machine or human feedback is best suited.

This feedback requires modeling of the learner and the different tutors. It is a complex system whose main characteristic is the number of dynamic data to be modeled and interpreted in order to provide answers to learners [17]. We have chosen a modeling by Multi-Agent Systems (MAS) to take into account the dynamic aspect of learner traces [62]. Agents are autonomous, problems solving computational entities capable of effectively performing operations in dynamic unpredictable environments. Agents interact and maybe cooperate with other agents. They are capable of exercising control over their actions and interactions

Our feedback is based on the principle of analogy and machine learning. Our method is based on Case Based Reasoning (CBR) [1] [34]. It is a reasoning by analogy based on the following assumption: « if a situation A is similar to a situation B then the consequences of situation A will be adaptable to those of situation B. »

IV. Generic intelligent tutoring Architecture

A. *Preamble*

There has been a great research effort in learning strategies to be integrate into ITS. Meyer [40] has used the analogy to teach a lesser-known domain from a more familiar one. The paradigm of case-based reasoning has also been potential solution for obtaining new incrementing knowledge. When various strategies are implemented together in an ITS, as for instance in, the system selects the most appropriate one for the activity that the student is performing.

On the other hand, agent technology has been suggested by experts to be a promising approach to address the challenges of ITS.

B. *Multi-Agent System (MAS)*

Denning [21] considers that the fundamental question posed in computer science is: what can be automated efficiently? This leads to distinguish three main categories of problems.

1. Problems that cannot be solved with a computer, such as predicting the next lotto draw (numbers drawn).
2. Problems that are easy to solve with a computer, such as finding out whether a given whole number exists in a few billion integers. In this category of problems, there is only one possible answer or an optimal solution.
3. Problems where it is possible to use a computer to obtain a correct solution, not necessarily the best.

The aim of artificial intelligence is to provide a correct solution to complex problems. A correct solution depends on the quality of the information used as input and the structures and organization of the data chosen according to the characteristics of the problems to be solved.

Knuth [33] defines a data structure as an array comprising structural relations and whose processing is done by algorithms for accessing and / or modifying the structure [42]. In the field of education, intelligent tutoring systems are complex systems whose main characteristic is the number of dynamic data to be modeled and interpreted in order to provide answers to learners [17]. In agreement with Wooldridge [62], to take into account the dynamic aspect of the problem, we have chosen a Multi-Agent System for the organization of the data.

Multi-Agent Systems refer to models of cognitive psychology but they have the ambition to simulate the cognitive behaviors of computer «agents» (perception, situation scripts, «mental» representation, learning action schemes, etc.) An «agent» is an autonomous and communicating entity, capable of making decisions on its own according to its objectives and limited information about its environment. A Multi-Agent System then links (passive) objects and these (active) agents, the latter being able, depending on their roles and a set of operators, to modify the passive objects.

Agents can be defined as autonomous, problem solving computational entities capable of effectively performing operations in dynamic unpredictable environments. Any agent, in accordance with this definition, satisfies four properties: autonomy, social ability, reactivity and pro-activeness. By using intelligent agents in an ITS architecture, it is possible to obtain an individual tutoring system adapted to the needs and characteristics of every student [20].

C. *Case Based Reasoning (CBR)*

Case Based Reasoning [34] is presented as a methodology of reasoning by analogy and also a methodology of machine learning from the field of AI, able to use the specific knowledge of past experiences, formalized in the form of concrete problematic situations called cases. This problem-solving technique has its origins in psychological models of memory and human expertise.

In general, what motivates the use of CBR is the lack of a comprehensive theory of problem solving in the application domain. Such a theory would systematically indicate how to solve any problem. Sometimes there may be theories but they are not enough. In addition, one must have particular experience of problem solving, which makes an application of CBR possible.

Classically, CBR is seen as a mode of problem solving relying on the reuse of solutions of already solved problems. For a CBR application, we consider the concepts of problem and solution: solving a problem means associating a solution with it. A case is the representation of a problem-solving episode. It encodes a problem and a solution for that problem, usually with information about the relationship between this problem and this solution attached. The set of cases that a CBR system has available is called the case base, and one case from this base is called the source case (the problem part is called the source problem). Reasoning from cases is to solve a problem, called a target problem, based on a case basis and, in general, on knowledge specific to the field of application. One important feature of CBR is that of machine learning. It allows you to update existing cases or learn new cases.

The CBR cycle has several phases [1] (Fig. 5):

1. Retrieve the most similar case(s) to the new case.
2. Adapt or Reuse the information and knowledge in that case to solve the new case. The selected best case has to be adapted when it does not match perfectly the new case.
3. Evaluate or Revise of the proposed solution.
4. Learn or Retain the parts of this experience likely to be useful for future problem solving.

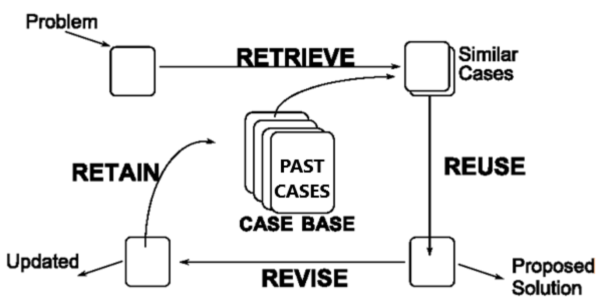

Fig.5.Cycle of CBR

To reason by analogy, a CBR system must have a case base. It may also need knowledge for the different stages of reasoning, in particular, similarity (or remembering knowledge, which is not necessarily reduced to a measure of similarity), adaptation knowledge, etc. In addition, domain knowledge (all available knowledge of the application domain) is often used in reasoning.

CBR deals with complex problems to convey plausible and reliable reasoning. It has been affected by other fields, for example, knowledge-based systems, machine learning, cognitive science, information retrieval, networking. neurons and fuzzy logic [52].

CBR systems produce solutions to the new problems by using the knowledge of past experiences. It can use solutions that are stored in its case base to solve similar repetitive problems which reduce the solution cost. If there is no similar case in the case base, generating and verifying the solution in the revision step is established based on the experiences of the existing case which presents a starting milestone instead of having to start from scratch, especially in applications where data are too scanty. Knowledge growth in CBR supports incremental learning.

Unlike other AI techniques, CBR system's expertise resides in the case base and the general knowledge, rather than being encoded in the form of rules. This helps the knowledge and the reasoning process to be implicit in the solution or explicitly recorded as a reused component [1].

In our context, CBR allows the problem-solving capacity of man to be synergized with the capacity of the computer system. The memory of both is mutually reinforcing to participate in problem solving. The use of a human tutor (peer/teacher) requires detecting that the learner is in a situation such that only the intervention of a human tutor is necessary. The aim is to detect, during the learning activity, behaviors likely to present pedagogical risks, to identify them in relation to existing cases and then to determine whether machine or human feedback is best suited to provide the necessary feedback.

D. *Multi-Agent CBR*

In many situations, AI techniques applied to a particular system do not need to be exclusive alternatives to each other, but can be viewed as complementary tools that can be grouped together within a single system. In addition, there is a wide variety of combination techniques which can be applied to particular problems. CBR is an open field of integration and combination of various types of techniques. CBR has the flexibility to combine effectively with other AI techniques [1] [47]

We place ourselves in the context of the development of dynamic and reactive systems, capable of adapting quickly and progressively to changes in the needs and uses of their users. In classical CBR, the ability of the system to adapt is limited by the fact that knowledge models and reasoning mechanisms are defined at the design stage and are therefore very difficult to evolve.

This article deals with a Multi-Agent CBR, a reasoning which exploits the traces of interaction, left by the learner, recovered from the LMS. Compared to conventional CBR, the Multi-Agent CBR we propose aims at the same principle: to recover past experience and then adapt it to provide a solution to the current problem. In practice, however, the mechanisms used are different. In our context, reasoning can no longer be seen as a cycle consisting of five successive and identifiable steps. On the contrary, the steps are intertwined and the back and forth between the steps multiply in order to clarify the description of the problem and its resolution.

The combination of the two technologies CBR and MAS has been proposed [22]. However, a major problem for these systems is the difficulty of adapting and evaluating the proposed solution.

[26] proposes the integration of agent technology and CBR to develop an ITS. This work includes a solution for adapting and evaluating the proposed solution. However, a major problem for this system is that it does not integrate, in the learning process, human tutoring (peers and teachers) and its articulation with machine tutoring.

V. CBR-MAS: ARCHITECTURE

A. *Architecture multilayer*

We present the CBR-MAS architecture for enhancing the intelligent learning environment. Our general architecture is formed by the three components that generally characterize an ITS. The architecture we propose is based on the structure of Woolf [63]. It consists of the following three components (Fig.6):

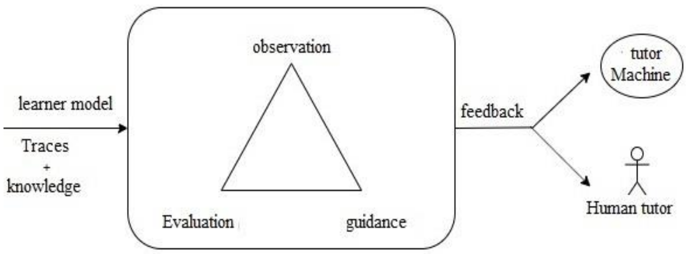

Fig. 6. Intelligent Tutoring Module.

1. Domain model. To ensure the genericity and reusability of our system, we use ontologies that offer a solution to manage heterogeneity. Ontology allows us to explicitly link pedagogical strategies to different cognitive and pedagogical theories [28]. Our approach separates the representation of knowledge, by its domain-specific nature, and its processing. This makes it possible to have a module which is independent of the domain to be taught. In other words, our module will apply to different domains such as the teaching of humanities, languages or engineering sciences.
2. Learner model and tutor model. We believe that these two models are strongly linked and therefore we propose for their modeling to implement a dynamic Multi-Agent CBR.
3. Interface. These are the graphical and adaptive interfaces for the different actors/users of the module.

By interacting with LMS, the learner produces traces that are digital fingerprinting from his or her own experiences. The trace is the central object of our approach. A trace represents the result of tracing the learner's interactions with LMS. The interaction traces allow the learner's problem-solving experiences to be memorized and thus reused. Traces of interaction are also used as sources of knowledge to generate other knowledge useful to the reasoning process.

Past experiences, which we will call «episodes», are remembered when there are similarities. This mechanism guarantees the flexibility and adaptability of the reasoning process. Episodes are always linked to the traces that contain them. Therefore, at any moment, it is possible to find indicators linked to the current episode and to use them to feed the reasoning process.

The Multi-Agent CBR takes into account the evolution and dynamic of the educational path to be analyzed. The analysis is based on the continuous comparison that the system will make between the learner's activity and the traces stored in the case base. These traces are described by the set of indicators that determine the course of learning. A determining indicator is a fact which played an effective role in the way in which the events unfolded.

We present here an approach and a Multi-Agent CBR developed within the framework of a generic ITS. The Multi-Agent CBR model will apply to any area where the target problem is dynamic.

When designing a Multi-Agent System, you must define the objectives of the agents and the tasks that must be carried out to achieve those objectives. In our CBR-MAS system, the main objective is to help the learner during his learning process. This is to identify the profile and needs of the learner to provide personalized learning. In order to achieve these goals, agents must perform the following main tasks: (1) to monitor the learner's activities, (2) to generate the student model, (3) to retrieve similar cases (4) to adapt solutions. Agents will go through different steps, in that order and allowing themselves to backtrack.

The architecture is based on 4 levels of agents leading to a pyramidal relationship (Fig. 7). Within this approach, the CBR-MAS architecture consists of the following layers: (1) The lowest level allows the dynamic and incremental generation of the target case (learner model), (2) The second level implements a dynamic and incremental retrieve process, (3) The third level is in charge of giving feedback, (4) The last level is in charge to enrich the base of cases.

Our CBR-MAS is composed of intelligent agents working to find the most similar cases. Cooperation between the various agents will make it possible to achieve the goal of offering the learner the feedback appropriate to his profile. This feedback can be machine as well as human.

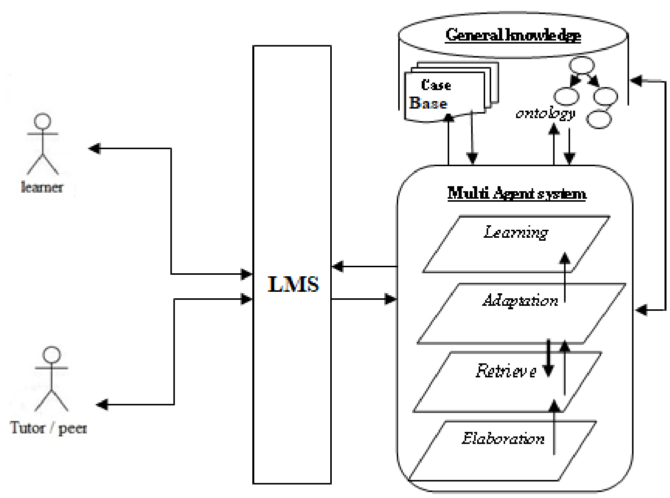

Fig. 7. Intelligent Tutoring Module: CBR Multi-Agent.

From the point of view of the CBR-MAS, the learners' models of the ITS architecture correspond to cases which are taken care of by agents. With a dynamic CBR-MAS cycle and an updated base of cases, the system adapts to changes in the environment and therefore the feedback is also adaptive.

We use the traces to model the learner. To provide effective learning, it is necessary to take into account all of the learner's traces, analyze these traces and finally provide feedback. Thus, the learner's modelling is done continuously and incrementally, the analysis and the feedback are continuous. By using agents, we respond to the dynamic nature of CBR. In the following sections, we describe the different levels of agents. Each level represents a stage in the dynamic CBR cycle.

B. Elaboration Layer

Agents for the dynamic and incremental elaboration of the target case (elaboration level). It is about constantly modeling the knowledge of the learner. The objective is to constantly represent the state of the learner (action and feedback). This state constitutes the learner's model which describes his acquired knowledge in the area to be learned. The agents of this layer follow the learner during his use of the LMS and directly and permanently retrieve the traces of his activities. Traces offer the possibility to build dynamically new case. The learning model includes several courses and courses which will allow a good description of the condition and subsequently a good diagnosis.

The role of the agents is to convert the traces retrieved from the LMS into a target case. A case is the basic component of a case-based system. In the literature, there is no consensual definition of the term case. Indeed, its definition is linked with respect to its representation format [34]. Traditionally, there have been several types of methods for representing cases: structural, textual, conversational, and hybrid.

- Structural format: A case is represented according to a common structure. It is built from the extraction of important features that model the problem to be solved. Knowledge representation formalism can be static, eg frames [46], description logic [23] or dynamic, eg objects [10].
- Textual format: A case can be represented in unstructured form (free text) or represented in semi-structured form, a text divided into several portions labeled by descriptors [38].
- Conversational format: A case is represented through three parts (description, questions, action). The description describes the problem to be solved in text form. The questions represent a series of questions and answers. Each question has a weight to represent its importance to the case. The action is a textual description of the solution to be implemented [2].
- Hybrid format: A case is represented by combining two or more formats.

Thus, we can see that the choice of the representation format is a problem that arises when creating a CBR. Indeed, this important step can influence all the other phases of the CBR cycle. Consequently, since we made the choice to graft our intelligent tutoring module to existing LMSs, we decided to retrieve the learner's traces in XML (Extensible Markup Language) [29] format and that the agents convert them. in target case using Case Markup Language (CaseML) [16].

CaseML is a standard vocabulary for describing cases for distributed case-based reasoning. A case mainly consists of three parts. The description of the problem that contains the values of the attributes that trigger the feedback. The solution that contains the description of the feedback to be performed. The relationship, optional part, describes the links between the cases. Multiple cases can be used to represent a single situation.

C. Retrieve Layer

In this phase, the system searches for a similar resolved case by comparing the new cases with the existing case base. The quality of the adaptation depends on the quality of the retrieve. In our context, several cases can be recalled in order to solve a single target problem. This can be done in at least two ways:

1. Either the recall module selects several source cases then, a module for combining these source cases makes it possible to build a solution of the target problem.
2. Either, the recall module selects a single source case which will give a partial solution to the target problem, then, the recall selects a new source case which will contribute to complete the solution, and so on, until having a complete solution.

Agents look for similar solutions by matching the source and target problems. These agents will use both the "Lazy Induction of Descriptions" (LID) [6] classification algorithm and dynamic clustering techniques.

The goal of LID is to classify a problem as belonging to one of the classes of solutions. The main idea of the LID is to determine what are the most relevant characteristics of the problem and to search the case base for cases sharing these relevant characteristics. The problem is classified when LID finds a set of relevant characteristics shared by a subset of cases all belonging to the same class of solution. Then the problem is classified in this class of solution. The LID is based on two main notions: 1) similarity is constructed as a symbolic description of what is shared between the previous cases and a specific problem to be classified, and 2) there is an evaluation function to help the system to decide which relationships between attributes are relevant to share with previous cases.

The traces retrieved from the LMS are constantly evolving and therefore also the target cases: this is the problem of data evolution. Moreover, new traces may appear with their own characteristics: this is the problem of the arrival of new data. The goal is for the grouping of characteristics into clusters to be relevant at all times. The characteristics of these cases are the clustering data, the evolution and arrival of these new cases will therefore modify the groupings made previously. This is indeed dynamic and incremental clustering [15] [44].

D. Adaptation Layer

Two critical stages in case-based design are the retrieval and the adaptation. The purpose of adaptation is to build on the remembered source case to solve the target problem, often by modifying the solution associated with the source case by relying on the difference between the source problem and the target problem. In general, adaptation consists in solving a reasoning problem by analogy: knowing the source and target problems, the links between these two problems (similarities, differences) and the links between the source problem and its solution (problem solving), we try to establish a target solution. It can be manual, copying, or automatic using algorithms, formulas and rules [61].

This level is responsible for adapting the solution of the case or similar cases selected by the previous level. The adaptation process can be as simple as the substitution of a component of the recovered solution or as complex as a complete change in the structure of the solution. It is about giving personalized feedback to the learner. Feedback can be machine or human (peers or teacher).

We suggest compositional adaptation, because many cases at the same time can be similar to the target case, and in this way, there will be the possibility of combining the corresponding solutions in an efficient way giving the final solution. In compositional adaptation, the solutions of several cases are combined to produce a new composite solution [61] [38]. If the solutions of the case or cases identified recommend machine feedback then the agents will use the compositional adaptation to propose solutions adapted to the learner. Otherwise the agents will offer the learner a list of peers, more advanced learners than him, or put him in direct contact with the teacher in charge of the course.

E. Learning Layer

Unlike expert systems which require relatively exhaustive modeling of the world you want to reason about, CBR systems memorize problem-solving experiences as you go. Admittedly, at start-up, the system is less efficient, but it gains in skills as it goes, and it is less difficult to maintain.

Retrieval phase is often based on an organization of the case base by an index hierarchy and / or a measure of similarity / dissimilarity. Indexing generally aims to generalize (and/or abstract) the source problem with a view to solving the problem that led to the solution of the source case.

One of the reasons for developing a CBR system is that these systems are able to 'learn' continuously from experience, not only memorizing the problems solved but also refining the domain model [37] [19], and require experts in the field to describe these experiences in such a way that they are reusable. This step completes the experience feedback loop that is a necessary prerequisite for enabling a system to learn from experiences. This phase consists of memorizing, if deemed appropriate, the case formed by the target problem and its solution.

Machine learning methods can be used in order to improve the base of cases (adding, creating, deleting cases) of the similarity measure (adjusting weights) and of the solution transformation (new adaptation rules), as well as, techniques from statistics and information theory.

VI. IMPLEMENTATION AND FUTURE WORKS

The complexity of the field prompted us to make a choice of progressive implementation. Now our system is called Intelligent Multidisciplinary Tutorial System (STIM). The collection and representation of traces, the entry point, is essential for the effective functioning of the system.

We chose to start with the implementation of a minimalist interface. This interface contains several multidisciplinary courses. It allows the learner to take a course, take a quiz and check their results. The purpose of this step is to recover traces of the learner's activity. Indeed, a set of information is extracted. The information retrieved is multiple:

- • Direct information: validation of a chapter of the course, of all the chapters, of a quiz, ...
- • Indirect information: movement and clicks of the mouse, time spent on a chapter, a course, a quiz, etc.

The data collected is structured and saved in an XML file. Before starting the implementation of our dynamic CBR agent model, we also developed an "Analysis / Decision" module, using a static CBR. This additional module allows you to test different pedagogical strategies. It compares the data in the XML file with an experimental case base. The learner is classified among four predefined categories (good / average / to monitor/ poor). The situation "to monitor" requires the attention of the human tutor, and "poor" his direct intervention. This static CBR, allowed us to validate the phase of recovery and representation of traces. The work in progress is to implement the dynamic version of STIM. The aim is to develop the different levels of the MAS (Fig. 7) which respectively represent the different stages of dynamic CBR.

VII. CONCLUSION

This paper presents an approach that aims to reduce the dropout or failure rate of students in universities. Dropouts mainly take place during the first cycle. It is therefore necessary to support students entering the University in a different way because it must be considered that the transition between high school and university remains difficult for the majority of new students. In addition, repeating or dropping out of a student increases the funding needed for their training.

Our system will allow teachers to better structure their lessons and focus on students in difficulty in order to better support them. It will facilitate the detection of points of difficulty in a personalized way for each student. The machine tutor will help the student with basic learning that does not require human intervention. The teacher will be able to find his or her true place in the learning process.

In this paper, we have presented the architecture of an intelligent tutoring module that enables LMSs to be equipped with a Machine/Human tutoring functionality. We use a dynamic CBR to evaluate the potential evolution of an observed situation. This architecture is based on 4 layers of agents having a pyramidal relationship and implementing dynamic and incremental CBR. The system is based on heterogeneous data.

The aim is to move from an exhaustive and factual description of the learner's state to a level of knowledge description that allows a synthetic characterization of this state. The continuous processing of information coming from the environment makes it possible to suggest feedback to the actors (students and tutors). To do this, the successful representation of the information must be formalized. In order to represent the learner's state, it is necessary to proceed with the construction of an ontology of the domain in order to be able to categorize the different indicators necessary and obligatory to follow the student's learning and to propose the adequate feedback.

We are in the context of the development of complex, dynamic and responsive systems, capable of adapting quickly, and gradually, to the changing needs and uses of their users. This point constitutes a first scientific problem.

The number of learners to take in undergraduate university courses can turn out to be considerable. However, current ITSs are often extremely limited in terms of the number of learners to support. This is why another scientific problem to study will be scaling up.

References

1. A. Aamodt. Knowledge-Intensive Case-Based Reasoning and Intelligent Tutoring. https://www.researchgate.net/publication/228601629_Knowledge-Intensive_Case-Based_Reasoning_and_Intelligent_Tutoring. 2009.
2. D. W. Aha, L. A. Breslow, H. Muñoz-Avila. Conversational case-based reasoning. Applied Intelligence., 2001, 14(1), pp. 9-32.
3. J. R. Anderson, E. Skwarecki. The automated tutoring of introductory computer programming. Common. ACM., 1986, 29(9), pp. 842–849.
4. C. Angelaki, I. Mavroidis. Communication and social presence: the impact on adult learners' emotions in distance learning. Eur.J. Open Distance E-learn., 2013, 16(1), pp. 78–93.
5. M. Arguedas, F. Xhafa, T. Daradoumis. An ontology about emotion awareness and affective feedback in elearning. In: International Conference on Intelligent Networking and Collaborative Systems., 2015, pp. 156–163, doi:10.1109/INCoS.2015. 78..
6. E. Armengol, E. Plaza. Lazy induction of descriptions for relational case-based learning. In European Conference on Machine Learning. Springer, Berlin, Heidelberg., 2001, pp. 13-24.
7. I. Arroyo, B. P. Woolf, W. Burelson, K. Muldner., D. Rai, M. Tai. A multimedia adaptive tutoring system for mathematics that addresses cognition, metacognition and affect. Int. J. Artif. Intell.Educ, 2014, 24(4), pp. 387–426, doi:10.1007/s40593-014-0023-y.
8. A. Balakrishnan. On modeling the affective effect on learning.In: Multi-disciplinary Trends in Artificial Intelligence, 2011, pp. 225–235, doi:10.1007/978-3-642-25725-4_20.
9. K. Barker, V. K. Chaudhri, S. Y. Chaw, P. Clark, J. Fan, D. J. Israel, S. Mishra, B. W. Porter, P. Romero, D. Tecuci, and others. A Question-Answering System for AP Chemistry: Assessing KR&R Technologies.," in KR, 2004, pp. 488–497.
10. R. Bergmann. Experience management: foundations, development methodology, and internet-based applications, 2002, Springer-Verlag.
11. F. Bertola, V. Patti. Ontology-based affective models to organize artworks in the social semantic web. Inf. Process. Manag., 2016, 52(1), pp. 139–162, doi: 10.1016/j.ipm.2015.10.003.
12. B. Bloom. Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain, 1984, 2nd edn. Addison Wesley, Reading.
13. B. Bourdeau,,M. Grandbastien. Modeling Tutoring Knowledge J. Advances in Intelligent Tutoring Systems, Springer, Studies in Computational Intelligence, 2010, volume 308, pp.123-143
14. [B. P. Butz, M. Duarte, and S. M. Miller. An intelligent tutoring system for circuit analysis. IEEE Trans. Educ., 2006, 49(2), pp. 216–223.
15. M. Charikar, C. Chekuri, T. Feder, R. Motwani. Incremental clustering and dynamic information retrieval. SIAM Journal on Computing, 33(6), 2004, pp. 1417-1440.
16. H. Chen, Z. Wu. On case-based knowledge sharing in semantic web. 15th International Conference on Tools with Artificial Intelligence, 2006 pp. 200– 206.
17. M. Clemens. The art of complex problem solving, http://www.idiagram.com/CP/cpprocess.html, 2005.
18. P. Cohen, J. Kulik, C. Kulik. Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal, 1982, 2 (19), pp. 237-248.
19. A. Cordier, B. Fuchs, J. Lieber, A. Mille. Failure Analysis for Domain Knowledge Acquisition in a Knowledge-Intensive CBR System. In: Case-Based Reasoning Research and Development, Proceedings of the 7th International Conference on Case Based Reasoning, ICCBR 2007, Belfast, Northern Ireland, 2007, pp. 463-477.
20. X. Daomin. Multi-Agent Based e-Learning Intelligent Tutoring System for Supporting Adaptive Learning. Fourth International Conference on Intelligent Systems Design and Engineering Applications, 2014, pp. 393-397.
21. P. J. Denning. Computer science: the discipline. in Anthony Ralston & David Hemmindinger (eds.), Encyclopedia of Computer Science, 2000, 32, no 1, pp. 9-23.
22. P.Dhyani, O.P. Rishi. Agent-based distributed intelligent tutoring system using case-based reasoning. In Intelligent Systems: Concepts, Methodologies, Tools, and Applications, IGI Global, 2018, pp 901-928.
23. F. M. Donini, M. Lenzerini, D. Nardi, A. Schaerf. Reasoning in description logics. Principles of Knowledge representation, 1996, pp. 191-236.
24. M. C. Duffy, R. Azevedo. Motivation matters: interactions between achievement goals and agent scaffolding for self-regulated learning within an intelligent tutoring system. Comput. Hum. Behav. 2015, 52, pp. 338–348. doi: 10.1016/j.chb.2015.05.041.
25. H. H. Emurian, H. K. Holden, R. A. Abarbanel. Managing programmed instruction and collaborative peer tutoring in the classroom: Applications in teaching JavaTM. Comput. Human Behav., 2008, 24(2), pp. 576–614.
26. C. González, J. C Burguillo, M. Llamas, & R. Laza. Designing intelligent tutoring systems: A personalization strategy using case-based reasoning and multi-agent systems. ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal, 2013, 2(1), pp. 41-54.
27. K. Hafner. Software tutors offer help and customized hints", https://www.nytimes.com/2004/09/16/technology/circuits/16tuto.html, 2004.
28. Y. Hayashi, J. Bourdeau, R. Mizoguchi. Using Ontological Engineering to Organize Learning/Instructional. Theories and Build a Theory-Aware Authoring System, IJAIED, 2009, 19 (2), pp. 211-252.
29. C. Hayes, P. Cunningham. Shaping a CBR View with XML in Case-Based Reasoning Research and Development. Proceedings of the Third International Conference on Case-Based Reasoning, 2000, pp. 468-481.
30. N. Jaques, C. Conati, J. Harley, R. Azevedo. Predicting Affect from Gaze Data During Interaction with an Intelligent Tutoring System. In: 12th International Conference, 2014, pp. 29–38, doi:10.1007/978-3-319-07221-04.
31. S. Jiménez, R. Juárez-Ramírez, V. H. Castillo, A. Noriega. Integrating affective learning into intelligent tutoring systems. Universal Access in the Information Society, 2018, 17(4), pp. 679-692.
32. R. John. Canvas LMS course design: design, build, and teach your very own online course using the powerful tools of the Canvas Learning Management System. Editor. 2014, PACKT Publishing.
33. D. E. Knuth. The Art of Computer Programming: Fundamental Algorithms, 1997, 3rd ed., Addison Wesley.
34. J. Kolodner. Case-based reasoning. 1993, San Mateo, CA: Morgan Kaufmann Publishers.
35. U. Kose, D. Koc (Ed.). (2014). Artificial Intelligence applications in distance education. IGI Global. 2014
36. V. Krishnamoorthy, B. Appasamy, and C. Scaffidi. Using intelligent tutors to teach students how APIs are used for software engineering in practice, IEEE Trans. Educ., 2013, 56(3), pp. 355–363.
37. D. Leake, A. Kinley, D. Wilson. Learning to Integrate Multiple Knowledge Sources for Case-Based Reasoning. In: Proc. of the 15th International Joint Conference on Artificial Intelligence, 1997, pp. 246-251.
38. M. Lenz, B. Bartsch-Spörl, H. D. Burkhard, S. Wess. Case-based reasoning technology: from foundations to applications. Springer, 2003, volume 1400.
39. H. C. K Lin, C. H. Wu, Y. P. Hsueh. The influence of using affective tutoring system in accounting remedial instruction on learning performance and usability. Comput. Hum. Behav., 2014, 41, pp. 514–522.
40. C. Meyer. Hypermedia environment for learning concepts based on inter-domain analogies as an educational strategy. In: International Conference on Intelligent Tutoring Systems. Springer, Berlin, Heidelberg, 2002. pp. 281-290.
41. E. Mousavinasab, N. R. Zarifsanaiey, S. Niakan Kalhori, M. Rakhshan, L. Keikha, M. Ghazi Saeedi. Intelligent tutoring systems: a systematic review of characteristics, applications, and evaluation methods. Interactive Learning Environments, 2021, 29(1), 142-163.
42. M. Murr. How forensic tools recover digital evidence (data structures)," http://www.forensicblog.org/how-forensic-tools-recover-digital-evidence-data-structures/, 2007.

43. L. S. Myneni, N. H. Narayanan, S. Rebello, A. Rouinfar, and S. Pumtambekar, An interactive and intelligent learning system for physics education. IEEE Trans. Learn. Technol., 2013, 6(3), pp. 228–239.
44. M. Nentwig, E. Rahm. Incremental clustering on linked data. In IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 531-538.
45. R. Nkambou., J. Bourdeau, R. Mizoguchi. Advances in Intelligent Tutoring Systems. Springer, Heidelberg: Studies in Computational Intelligence, 2010, vol 308.
46. E. Plaza. Cases as terms: A feature term approach to the structured representation of cases. In International Conference on Case-Based Reasoning. Springer Berlin Heidelberg, 1995, pp. 265-276.
47. J. Prentzas, I. Hatzilygeroudis. Combinations of Case Based Reasoning with Other Intelligent Methods. International Journal of Hybrid Intelligent Systems, ISO Prss, 2009, vol. 6, No. 4, pp.189-209.
48. A. Ramírez-Noriega, R. Juárez-Ramírez, Y. Martínez-Ramírez. Evaluation module based on Bayesian networks to Intelligent Tutoring Systems. International Journal of Information Management, 2017, vol. 37, issue, part A, pp. 1488-1498.
49. L. Razzaq, J. Patvarczki, S. Almeida, M. Vartak, M. Feng, N. N. T. Heffernan, K. R. Koedinger. The ASSISTment Builder: Supporting the Life Cycle of Tutoring System Creation. IEEE Tr. on Learning Technologies, 2009, 2 (2), pp. 157-166.
50. U. D. C Rica., S. Pedro, M. D. Oca, C. Rica, C. The emotional intelligence, its importance in the learning process. Educacio'n, 2012, 36(1), pp. 1–24.
51. P. Richard, M. Gagnon, J. Fortuny, N. Leduc, M. Tessier-Baillargeon. Means of Choice for Interactive Management of Dynamic Geometry Problems Based on Instrumented Behaviour. In American Journal of Computational Mathematics, 2013, vol 03, pp. 41-51.
52. S. Sathasivam. Applying Knowledge Reasoning Techniques in Neural Networks. Australian Journal of Basic and Applied Sciences, 2012, vol. 6, No. 2, pp. 53-56.
53. S. Suebnukarn and P. Haddawy. COMET: A collaborative tutoring system for medical problem-based learning. IEEE Intell. Syst., 2007, 22(4), pp. 70–77.
54. J. Tchetagni, R. Nkambou, J. Bourdeau. Explicit Reflection in Prolog-Tutor, IJAIED, 2007, 17(2), pp. 169-217.
55. M. Tsuei. Using synchronous peer tutoring system to promote elementary students' learning in mathematics. Comput. Educ., 2012, 58(4), pp. 1171–1182.
56. B. Toussaint, V. Luengo, L. Vadcard, J. Tonetti. Apprentissage de la chirurgie orthopédique assisté par ordinateur : Le cas du Système Tutoriel Intelligent TELEOS. Field Actions Science Reports. The journal of field actions, 2014.
57. B. Vesin, M. Ivanović, A. KlašNja-MilićEvić, Z. Budimac. Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. Expert Syst. Appl., 2012, 39(15), pp. 12229–12246.
58. D. Xu, S. S. Jaggars. Performance Gaps Between Online and Face-to-Face Courses: Differences Across Types of Students and Academic Subject Areas. The Journal of Higher Education, 2015, 85(5), pp. 633-659., doi:10.1353/jhe.2014.0028.
59. D. Wang, H. Han, Z. Zhan, J. Xu, Q. Liu, G. Ren. A problem solving oriented intelligent tutoring system to improve students' acquisition of basic computer skills. Published in Computer & Education, 2015, volume 81, pp. 102-112.
60. E. Wenger. Artificial Intelligence and Tutoring Systems: computational and cognitive approaches to the communication of knowledge.. Morgan Kaufman Publishers, 2014.
61. W. Wilke, R. Bergmann. Techniques and Knowledge Used for Adaptation During Case Based Problem Solving. Tasks and Methods in Applied Artificial Intelligence, LNAI 1416, Springer-Verlag, 1998, pp. 497-505.
62. M. Wooldridge. An Introduction to Multi-Agent Systems, 2nd ed., Wiley, 2009.
63. B. P. Woolf. Building Intelligent Interactive Tutors. Morgan Kaufmann., 2009.