

Our System IDCBR-MAS: from the Modelisation by AUML to the Implementation under JADE Platform

Abdelhamid Zouhair¹, El Mokhtar En-Naimi¹, Benaïssa Amami¹, Hadhoum Boukachour², Patrick Person², Cyrille Bertelle²

¹*LIST Lab, The University of Abdelmalek Essaâdi, FST of Tangier, Morocco*

²*LITIS Lab, The University of Le Havre, France*

Abstract — This paper presents our work in the field of Intelligent Tutoring System (ITS), in fact there is still the problem of knowing how to ensure an individualized and continuous learners follow-up during learning process, indeed among the numerous methods proposed, very few systems concentrate on a real time learners follow-up. Our work in this field develops the design and implementation of a Multi-Agents System Based on Dynamic Case Based Reasoning which can initiate learning and provide an individualized follow-up of learner. This approach involves 1) the use of Dynamic Case Based Reasoning to retrieve the past experiences that are similar to the learner's traces (traces in progress), and 2) the use of Multi-Agents System. Our Work focuses on the use of the learner traces. When interacting with the platform, every learner leaves his/her traces on the machine. The traces are stored in database, this operation enriches collective past experiences. The traces left by the learner during the learning session evolve dynamically over time; the case-based reasoning must take into account this evolution in an incremental way. In other words, we do not consider each evolution of the traces as a new target, so the use of classical cycle Case Based reasoning in this case is insufficient and inadequate. In order to solve this problem, we propose a dynamic retrieving method based on a complementary similarity measure, named Inverse Longest Common Sub-Sequence (ILCSS). Through monitoring, comparing and analyzing these traces, the system keeps a constant intelligent watch on the platform, and therefore it detects the difficulties hindering progress, and it avoids possible dropping out. The system can support any learning subject. To help and guide the learner, the system is equipped with combined virtual and human tutors.

Keywords — Intelligent Tutoring Systems (ITS), Multi-Agents System (MAS), Incremental Dynamic Case Based Reasoning (IDCBR), Similarity Measure, Traces.

I. INTRODUCTION

E-LEARNING is a computer system which offers learners another means of learning. Indeed it allows learner to break free from the constraints of time and place of training. They are due to the learners' availability. In addition, the

instructor is not physically present and training usually happens asynchronously. However, most E-learning platforms allow the transfer of knowledge in digital format, without integrating the latest teaching approach in the field of education (e. g. constructivism [23], ...). Consequently, in most cases distance learning systems degenerate into tools for downloading courses in different formats (pdf, word ...). These platforms also cause significant overload and cognitive disorientation for learners. Today, it is therefore necessary to design and implement a computer system (i. e. intelligent tutor) able to initiate the learning and provide an individualized monitoring of the learner, who thus becomes the pilot of training. The system will also respond to the learner's specific needs.

Solving these problems involves first, to understand the behaviour of the learner, or group of learners, who use platform to identify the causes of problems or difficulties which a learner can encounter. This can be accomplished while leaning on the traces of interactions of the learner with the platform, which include history, chronology of interactions and productions left by the learner during his/her learning process. This will allow us the reconstruction of perception elements of the activity performed by the learner.

We consider an Intelligent Tutoring System (ITS), that is able to represent, follow and analyze the evolution of a learning situation through the exploitation and the treatment of the traces left by the learner during his/her learning on the platform. This system is based, firstly on the traces to feed the system and secondly on the reconciliation between the course of the learner (traces in progress) and past courses (or past traces). The past traces are stored in a database. Our system is able to represent, follow and analyze the evolution of a learning situation through the exploitation and the treatment of the traces left by the learner during his/her learning on the platform. The analysis of the course must be executed continuously and in real time which leads us to choose a Multi-Agents architecture allowing the implementation of a dynamic case-based reasoning. Recently, several research works have been focused on the dynamic case based reasoning in order to push the limits of case based reasoning system static, reactive

and responsive to users. All these works are based on the observation that the current tools are limited in capabilities, and are not able of evolving to fit the non-anticipated or emerging needs. Indeed the reuse of past experiences causes several problems, such as:

- Modeling: formalization of experience acquired (cases), indeed a few CBR systems are able to change over time the way of representing a case [6]. According Alain Mille, a case has to describe its context of use, which is very difficult to decide before any reuse and can change in time [22].
- Treatment: the use of the classic reasoning cycle is insufficient and inadequate in dynamic or emerging situations, unknown in advance.

In order to deal with this issue, we propose a Dynamic Case Based Reasoning based on a dynamic retrieve method, and we propose a dynamic retrieving method based on a complementary similarity measure, named Inverse Longest Common Sub-Sequence (ILCSS).

The rest of this paper is organized as follows: In the second section, we present a general introduction of intelligent tutoring system. In the third section we present a Multi-agents Case-Based Reasoning, and in the following part, we will propose the description of our approach in Case Based Reasoning and intelligent tutoring systems field: Incremental Dynamic Case-Based Reasoning founded on Multi-Agents System. In the next section, we present some development results of our system. Finally, we present a comparison between IDCBR-MAS system and other CBR system, and we will give the conclusion and our future work.

II. INTELLIGENT TUTORING SYSTEMS

Intelligent Tutoring Systems (ITS) are computer systems designed to assist and facilitate the task of learning for the learner. It can personalize learning for learners, providing a less expensive solution for a diverse generation of learners. They have expertise in so far as they know the domain knowledge, how to teach (pedagogical knowledge) and also how to acquire information about the learner. We note that, the general architecture of Intelligent Tutoring Systems was represented in our articles [10]. Many researches have been designed and implemented in Intelligent Tutoring System, in order to assist a learner in his/her/their learning. There are, for example, tutors or teaching agents who accompany learners by proposing remedial activities [11]. There are also the agents of support to the group collaboration in the learning [7] encouraging, the learners participation and facilitating discussion between them. Other solutions are based on Multi-Agents System that incorporate and seek to make cooperation among various Intelligent Tutoring System [5]. The Baghera platform [32] exploits the concepts and methods of Multi-Agents approach. Baghera assists learners in their work solving exercise in geometry. They can interact with other learners or teachers. The teachers can know the progress of the learners work in order to intervene if needed. These tools of distance learning do not allow an individualized, continuous

and real-time learners follow-up. They adopt a traditional pedagogical approach (behaviorist) instead of integrating the latest teaching approaches (constructivism and social constructivism [23], [30]). Finally, given the large number of learners who leave their training, the adaptation of learning according to the learners profile has become indispensable today.

Our contribution in these important areas is to design and develop an adaptable system that can ensure an automatic and a continuous monitoring of the learner. Moreover, our system is open, scalable and generic to support any learning subject.

III. MULTI-AGENTS CASE-BASED REASONING

A. Case-Based Reasoning

Case-Based Reasoning (CBR) is an artificial intelligence methodology which aims at solving new problems based on the solutions of similar past problems (past experiences) [14]. The solved problems are called source cases and are stored in a case-base (base of scenarios). The problem to be solved is called target case. A CBR is a combination of knowledge and processes to manage and re-use previous experience.

The Case-Based Reasoning cycle is composed of five steps as given at following figure (Fig. 1):

- Presentation: the current problem is identified and completed in such a way that it becomes compatible with the contents and retrieval methods of the case-base reasoning.
- Retrieve: The task of retrieve step is to find the most similar case or cases to the current problem in the case-base.
- Reuse: The goal of the reuse phase is to modify the solution of source case found in order to build a solution for the target case.
- Revise: The phase of revision is the step in which the solution suggested in the previous phase will be evaluated. If the solution is unsatisfactory, then it will be corrected.
- Retain: retaining the new experience and add it to the knowledge-base (case-base) [12], [1].

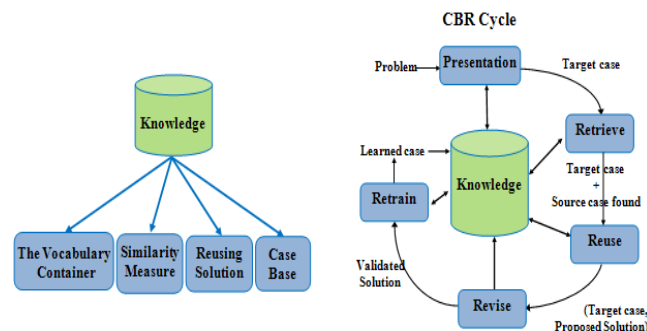


Fig. 1. The CBR components (Source [1], [12])

The systems based on the case-based reasoning can be classified into two categories [18]:

- Applications for static situation. For this type of system, the designer must have all the characteristics describing a case, in advance, in order to be able to realize its

model. A data model of the field is thus refined through an expertise in the field of application which can characterize a given situation. Thus, the cases are completely structured in this data model and often represented in a list (a: attributes, v: values). For example we have the system CHIEF [13] case base planner that builds new plans out of its memory of old ones. We do not exploit this type of CBR to develop our system, because in the approach for static situation, a problem must be completely described before starting the first step. Nevertheless in our situation, the learner traces (target case) evolve dynamically over time, so we must treat a dynamic situation with some particular features.

- Applications for dynamic situation. They differ when we compare them to static cases by the fact that they deal with temporal target cases (the situation), by looking for similar cases (better cases) based on a resemblance between histories (for more details on the subject, the reader may refer to [2], [18])). Several works relate to dynamic case based reasoning such as REBECAS [18] prediction of processes from observed behaviours, application to wildfire and SAPED [2].

B. Multi-Agents Case-Based Reasoning

The Multi-Agents System based on case based reasoning are used in many applications areas [25]. We can distinguish two types of applications (Table I):

- The Multi-Agents System in which each agent uses the case based reasoning internally for their own needs (level agent case based reasoning): This type is the first model that was applied in Multi-Agents CBR Systems. For this type of system, each agent is able to find similar cases to the target case in their own case base, also able to accomplish all steps of CBR cycle. For example we have the system ProCLAIM [29], MCBR [17] for distributed systems and CBR-TEAM [26] approach that uses a set of heterogeneous cooperative agents in a parametric design task (steam-condenser component design).
- The Multi-Agents System whose approach is a case based reasoning (level Multi-Agents Case Based Reasoning) : For this types of applications, the Multi-Agents Case Based Reasoning System distribute the some/all steps of the CBR cycle (Representation, Retrieve, Reuse, Revise, Retain) among several agents. The second category might be better than the first. Indeed the individual agents experience may be limited, therefore their Knowledge and prediction, so the agents are able to cooperate with other agents for a better prediction of the situation and they can benefit from the other agents capabilities. For example we have CCBR [21], RoBoCats [20] and S-MAS [24].

To our knowledge, no dynamic CBR cycle reasoning system exists.

We propose a system called Incremental Dynamic Case Based Reasoning-Multi-Agents System (IDCBR-MAS), able to find similar cases to the target case in their own case base. Our system is founded on 1) a dynamic cycle of case-based reasoning, and 2) a dynamic retrieving method based on a

complementary similarity measure, named Inverse Longest Common Sub-Sequence (ILCSS) (for more details on the subject, the reader may refer to [10, 34]).

IV. INCREMENTAL DYNAMIC CASE-BASED REASONING FOUNDED ON MULTI-AGENTS SYSTEM

A. General architecture of our approach IDCBR-MAS

Our problem is similar to the CBR for dynamic situation. Indeed, the traces left by the learner during the learning session evolve dynamically over time; the case-based reasoning must take into account this evolution in an incremental way. In other words, we do not consider each evolution of the traces as a new target. The intelligent system (IDCBR-MAS) which we propose offer important features:

- It is dynamic. Indeed we must continually acquire new knowledge to better reproduce human behaviour in each situation.
- It is incremental; this is its major feature because the trace evolves in a dynamic way for the same target case.

The main benefits of our approach are the distributed capabilities of the Multi-Agents System and the self-adaption ability to the changes that occur in each situation. The system that we propose consists of the three layers components (as indicated in Figure Fig. 2:

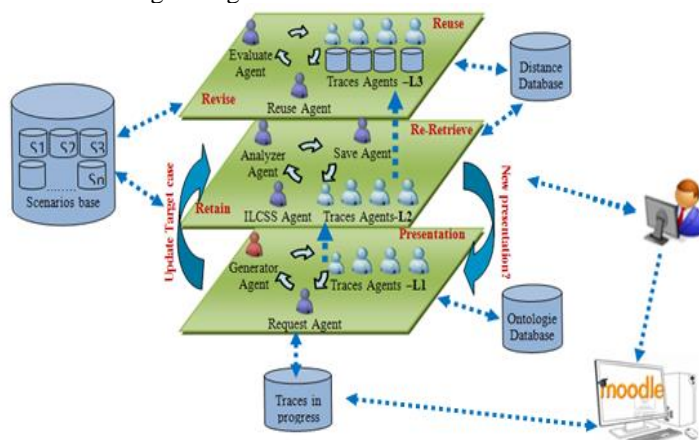


Fig. 2. General architecture of our approach

1) *Presentation layer*: The role of agents of this layer is to manage information arrived from the environment (the learner traces). This information feed the representation layer. The goal of this layer is to be both, a picture of the current situation being analyzed and to represent its dynamic evolution over time. The presentation layer contains the following agents:

- Request Agent: The role of this agent is to establish the link between the system and the environment. They feed the system with information from Distributed Information Systems (file traces). Also the goal of this agent is to check if there is any change in the traces file.
- Generator Agent: the role of this agent is to create and/or update the Traces Agents-L1: The Request Agent transmits the data received from environment to the Generator Agent. Two cases of figure are presented: if the Traces Agents-L1 (i) related to the learner i exist,

then this last will be updated, else the Generator Agent creates a new Traces Agents-L1 (i).

- Traces Agents-L1: For each Lerner *i* we have a represented Trace Agent-L1(*i*). These agents will encapsulate the original traces of learners.
- 2) *Interpretation Layer and storage:* A set of agents allows the comparison between the current situation and past situations stored in the memory (scenarios). The Interpretation Layer contains the following agents:
- Traces Agents- L2: These agents contain the same information and data that have in the Trace Agents-L1 of the first layer. They differ by an abstraction of the data, originally described and managed by the Trace Agents-L1, that make it comparable to the past experiences stored in the memory.
 - ILCSS Agent: The role of this agent is to evaluate in a continuously way the similarity between the current situation and past experiences based on the similarity measure ILCSS. The retrieve step of our system is based on this agent. The ILCSS Agent save the distances between the current situation and past experiences in Distance Table. It is responsible for reviewing these distances every time whenever necessary.
 - Analyzer Agent: The goal of this agent is to check in a dynamic way if there is any change or update in Trace Agent -L2 (with the arrival of new information and data from the environment), then the Analyzer Agent asks ILCSS Agent to update Distance Table each time they have a change in the Trace Agent -L2, if not they asks the Request Agent if there is any change in traces file.
- 3) *Prediction and Decision Layer:* The role of agents of this layer is to predict the current situation by reusing past experiences selected by second layer. The choice of similar past experiences is evaluated by this layer, so one of these scenarios will be proposed to the learner. The layer contains the following agents:
- Traces Agents-L3: At this stage of reasoning the system adds a pointer to each agent the Traces Agents-L2. So the Traces Agents-L3 is identical to Traces Agents-L2 with a small difference, in fact for each Traces Agents-L2 we associate a list of similar scenarios through a pointer to the list of similar past experiences. The advantage of a pointer is that the list is not exhaustive and it changes dynamically over time following the change of the learner traces.
 - Reuse Agent: The role of this agent is to predict future events of the situation by reusing the past experiences to the current situation.
 - Evaluate Agent: The role of this agent is to evaluate the solution proposed by the Reuse Agent and to ensure that the similarity between the current situation and scenarios chosen by the Prediction layer is sufficient.
 - Human Tutor or Human Agent: The human tutor is solicited if the system detects a learning situation requiring his intervention (failure to find one or more similar scenarios to the current situation).

B. From static to dynamic CBR cycle

We modify the CBR cycle in order to be able to handle

dynamic situations and therefore we propose changes in the order of steps and a large change in the content of the steps of this cycle. In our approach the evaluation of the similarity between the current situation and similar past situations is a process continues. The retrieve step of the CBR cycle (as indicated in figure Fig.3) must take into account the change in the current situation in a dynamic way (in real-time). Our system will be able to repeat the retrieve step following the change of the current situation or whenever necessary.

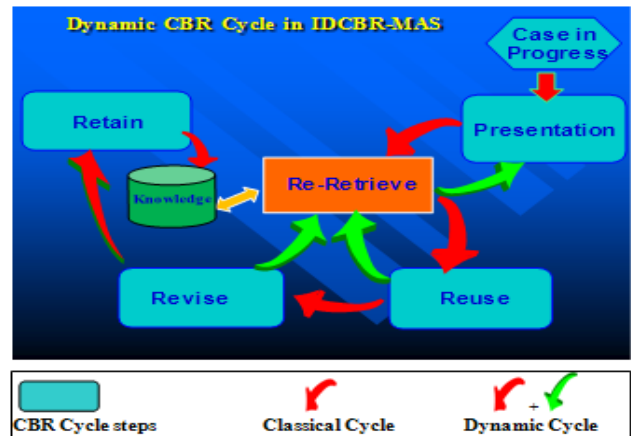


Fig. 3. Dynamic CBR Cycle in IDCBR-MAS

In addition, in our system the sequence of steps of the CBR cycle isn't important: in fact our system can stop each step in the CBR cycle and return to a previous step following the change of the current situation, and the order presentation – retrieve – reuse – revise – retain is not static or fixed, it can change and some steps can be re-run each time until the change in the situation.

Our agents are equipped with learning, communication and intelligence skills. They are able to stop the execution of the CBR cycle at a given step and time. They are able to re-run the different steps later following a change in the target situation. The highlight of our approach is that rerunning the retrieval step based our new dynamic similarity measure ILCSS. In each step CBR cycle of our approach we takes into account the previous results i.e. in time t_{i+1} we use the results in t_i . Therefore our CBR cycle takes into account the change of situation in a dynamic and incremental way.

1) *Retrieval steps* : Retrieval of previous case is one important step within the CBR paradigm. The success of retrieval step will depend on three factors: the case representation, case memory and similarity measure used to retrieve sources cases that are similar to the target case. There are two ways research for the sources case in dynamic situations:

- Research by evaluating similarity between the current situation and the already solved problems in a single dimension [18]. Several systems have been used this approach such as REBECAS [18] and SAPED [2].
- Research by evaluating similarity between the current problem and the already solved problems in a multiple dimension [2]. The multidimensional research, it is

realized in a single step by taking into account all the parameters describing the current problem at the same time. The multidimensional research is also used in several systems, such as CASEP2 [33].

2) *State of the Art on Similarity Measures*: Search for similar sources cases are based on the similarity measure. In this part, we present the principles similarity measures often used in case based reasoning, for more details on the subject, the reader may refer to [2] and to our articles [10], [34].

Biological Sequences Alignment: Dynamic Programming, is an important tool, which has been used for many applications in biology. It is a way of arranging the sequences of DNA, or protein to identify regions of similarity that may be a consequence of structural or functional relationships between the sequences. They are also used in different fields, such as natural language or data mining.

Minkowski distance: The Minkowski distance is a metric on Euclidean space which can be considered as a generalization of both the Euclidean distance.

Longest Common Sub-Sequence (LCSS): the goal is to find the longest subsequence common in two or more sequences [31]. The LCSS is usually defined as: Given two sequences, find the longest subsequence present in both of them. A subsequence is a sequence that appears in the same order, but not necessarily contiguous. The main goal is to count the number of pairs of points considered similar when browsing the two compared sequences.

There are other similarity measures such as **Dynamic Time Warping (DTW)**: The DTW algorithm is able to find the optimal alignment between two sequences. It is often used in speech recognition to determine if two waveforms represent the same spoken phrase. In addition to speech recognition, dynamic time warping has been successfully used in many other fields [2], such as robotics, data mining, and medicine.

TABLE I.
COMPARISON OF VARIOUS SIMILARITY MEASURES [2]

	Type	Dimension	Length
Biological Sequences Alignment	Symbolic	One-dimensional	Different
DTW	Digital	One-dimensional	Different
LCSS	Heterogeneous	Multidimensional	Different
Minkowski distance	Digital	One-dimensional	Same Length

3) *Inverse Longest Common Sub-Sequence*: The main goal of the retrieval phase in our system is to predict the behavior of the learner, by the reconciliation between the target trace and past traces or scenarios. The success of a case-based reasoning system depends primarily on the performance of the retrieval step used and, more particularly, on similarity measure used to



retrieve sources cases (scenarios) that are similar to the situation (traces in progress). Several research works have

been focused on the similarity measure. Furthermore, these methods are not well suited when we compare two dynamic and heterogeneous sequences. In order to deal with this issue, we propose a complementary similarity measure entitled **Inverse Longest Common Sub-Sequence** an extension of the Longest Common Sub-Sequence measure.

In our system IDCBR-MAS the target case or target trace can be represented as a various actions of the learner (learner traces). It can be represented also as a collection of semantic features $SF = (\text{object}, (\text{qualification}, \text{value}) +)$, we note $\text{object} = O$, $\text{qualification} = Q$ and $\text{value} = V$, $SF = (O, (Q, V) +)$, so the learner traces at time i , can be defined by the formula:

$$LT_i = \bigcup_{1 \leq k \leq i} SF_k$$

Where $SF_k = (O_k, (Q_{k,1}, V_1), \dots, (Q_{k,d}, V_d))$ is a sequence of $d+1$ dimension. Finally the learner traces at time $i+1$ is a multidimensional sequence.

Let A and B two Traces with size $n \times d$ and $m \times d$ respectively, where:

$$A = ((O_{A,1}, (Q_{A,1,1}, V_{A,1,1}), \dots, (Q_{A,1,d}, V_{A,1,d}), (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d}), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d})))$$

And

$$B = ((O_{B,1}, (Q_{B,1,1}, V_{B,1,1}), \dots, (Q_{B,1,d}, V_{B,1,d}), (O_{B,2}, (Q_{B,2,1}, V_{B,2,1}), \dots, (Q_{B,2,d}, V_{B,2,d}), \dots, (O_{B,m}, (Q_{B,m,1}, V_{B,m,1}), \dots, (Q_{B,m,d}, V_{B,m,d})))$$

For a Trace A , let $\text{Tail}(A)$ be the Trace:

$$\text{Tail}(A) = (O_{A,2}, (Q_{A,2,1}, V_{A,2,1}), \dots, (Q_{A,2,d}, V_{A,2,d}), \dots, (O_{A,n}, (Q_{A,n,1}, V_{A,n,1}), \dots, (Q_{A,n,d}, V_{A,n,d}))).$$

$\text{Tail}(A)$ is the trace A private their first vector.

The goal is to count the number of pairs vectors considered similar when compared through the two traces. The similarity between two vectors $(V_{A,i,1}, V_{A,i,2}, \dots, V_{A,i,d})$ from trace A , and $(V_{B,j,1}, V_{B,j,2}, \dots, V_{B,j,d})$ from trace B it determined according to a threshold δ : if for each $k = 1, \dots, d$ $|V_{A,i,k} - V_{B,j,k}| \leq \delta$. We also define an integer N , the parameter that will be able to control the temporal variance between two vectors of each of the traces in order to consider the two traces similar.

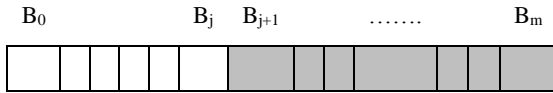
Let A and B two Traces, and given an integer N and a real number δ , we define the similarity measures between the two traces A and B , as follows recursive process: the process is initialized by comparing the two first vectors of traces (A, B) . If any of the two traces is empty then the value of the similarity measure is equal to 0, and the process stops. Else if any of the two vectors traces are similar, then the similarity measure in this case is "1" more the similarity between the two traces deprived of their first vectors. Else the similarity is equal to the maximum of the similarity between a trace and the other private its first vector.

At the instant $t=i+1$ the IDCBR-MAS system recovers the traces stored in the log file of server between the two instants t_i and t_{i+1} and we have $(A)_{t=i+1} = \text{Tail}(A)_{t=i+1} = A[t_i, t_{i+1}]$ see figure below.

$$\text{Tail}(A)_{t=i+1} = A_{[t_i, t_{i+1}]} = ((O_{A[t_i+1, t_i]}, (Q_{A[t_i+1, t_i], 1, 1}, V_{A[t_i+1, t_i], 1, 1}), \dots, (Q_{A[t_i+1, t_i], 1, d}, V_{A[t_i+1, t_i], 1, d}), \dots, (O_{A[t_i+1, t_i], n},$$

$(Q_{A[t_i+1,t_i],n,1}, V_{A[t_i+1,t_i],n,1}), \dots, (Q_{A[t_i+1,t_i],n',d}, V_{A[t_i+1,t_i],n',d}))$ avec $1 \leq i' \leq n'$

At the instant $t=t_i+1$ it only remains the block $B_{[j+1,m]}$ of the B traces (block of the trace B have not yet been compared with the target trace), where B_j describe, the last common element between the two traces (A) $t=t_i$ and B at the instant $t=t_i$.



$(B)_{t=t_i+1} = \text{Tail}(B)_{t=t_i+1} = B_{[j+1,m]}$
 $\text{Tail}(B)_{t=t_i+1} = B_{[j+1,m]} = ((O_{B_{[j+1,m]},1}, (Q_{B_{[j+1,m]},1,1}, V_{B_{[j+1,m]},1,1}), \dots, (Q_{B_{[j+1,m]},1,d}, V_{B_{[j+1,m]},1,d}), \dots, (O_{B_{[j+1,m]},m'}, (Q_{B_{[j+1,m]},m',1}, V_{B_{[j+1,m]},m',1}), \dots, (Q_{B_{[j+1,m]},m',d}, V_{B_{[j+1,m]},m',d})))$ with $1 \leq j' \leq m'$ ($m' = m-j$)

The measure between the target traces A and the source trace B at time $t = t_{i+1}$ will depend on the results instantly $t = t_i$ through the following recursive formula:

$$(ILCSS_{\delta,N,t_i+1}(A,B))_{t=t_i+1} = (ILCSS_{\delta,N,t_i}(A,B))_{t=t_i} + ILCSS_{\delta,N,t_i+1}(A_{[t_i,t_i+1]}, B_{[j+1,m]})$$

With

$$ILCSS_{\delta,N,t_i+1}(A_{[t_i,t_i+1]}, B_{[j+1,m]}) = \begin{cases} 0 \text{ if } A_{[t_i,t_i+1]} \text{ or } B_{[j+1,m]} \text{ is empty,} \\ 1 + ILCSS_{\delta,N,t_i+1}(A_{[t_i,t_i+1]}, B_{[j+1,m]}) \text{ if } O_{A_{[t_i,t_i+1]},i'} = O_{B_{[j+1,m]},j'}, Q_{A_{[t_i,t_i+1]},i',k'} = Q_{B_{[j+1,m]},j',k'} \\ |V_{A_{[t_i,t_i+1]},i',k'} - V_{B_{[j+1,m]},j',k'}| \leq \delta, \text{ with } 1 \leq i' \leq n', 1 \leq j' \leq m', |i' - j'| \leq N \text{ et } k' = 1 \dots d. \\ \max(ILCSS_{\delta,N,t_i+1}(\text{Tail}(A_{[t_i,t_i+1]}), B_{[j+1,m]}), ILCSS_{\delta,N,t_i+1}(A_{[t_i,t_i+1]}, \text{Tail}(B_{[j+1,m]}))) \text{ else.} \end{cases}$$

The index of the last common element between the two traces (A) $t=t_i+1$ and B at time $t = t_{i+1}$ is obtained using the following iterative formula.

J is initialized to 0;

And

$$j = j' \text{ if } O_{A_{[t_i,t_i+1]},i'} = O_{B_{[j+1,m]},j'}, Q_{A_{[t_i,t_i+1]},i',k'} = Q_{B_{[j+1,m]},j',k'} |V_{A_{[t_i,t_i+1]},i',k'} - V_{B_{[j+1,m]},j',k'}| < \delta \text{ and } j' > j$$

With $1 \leq i' \leq n', 1 \leq j' \leq m', |i' - j'| \leq N$ and $k'=1 \dots d$

We define the distance between the two Traces A and B as follow:

$$D_{\delta,N,t_i}(A,B) = 1 - \frac{ILCSS_{\delta,N,t_i}(A,B)}{\min(n,m)}$$

Where $D_{\delta,N,t_i}(A,B)$ verify the proprieties of the distance such

$$\text{as: } \begin{cases} D_{\delta,N,t_i}(A,B) \geq 0 \\ D_{\delta,N,t_i}(A,B) = 0 \text{ equal } A \approx B \\ D_{\delta,N,t_i}(A,B) = D_{\delta,N,t_i}(B,A) \end{cases}$$

$A \approx B$: A and B are two similar traces.

4) *Learner's Traces and case structure*: Based on the general definition of a trace given in [19], "a trace is a thing or a succession of things left by an unspecified action and relative to a being or an object; a succession of prints or marks which the passage of a being or an object leaves; it is what one recognizes that something existed; what remains of a past thing". In ITS literature, a digital trace is an observed

collection, all structured information resulting from an interaction observation temporally located [22].

In our context, a digital trace is resulting from an activity observation representing a process interactional signature. Indeed, it is composed of the objects which are respectively located the ones compared to the others when observed and registered on a support. That means that a trace is explicitly composed of the structured objects and registered compared to a time representation of the activity traces. The structuring can be sequentially explicit (each trace observed is followed and/or preceded by another) or can also come from the temporal characteristic of the objects traces [19]. Indeed, the structuring depends on the type of the time representation and the time of the activity traces. We can distinguish two types of representations:

- They can be a temporal interval determined by two dates, (start and end of observation). In this case, the observed traces may be associated with an instant or an interval of time. Then we will be able to take into account chronological relationships between observations';
- They can be a sequence of unspecified elements (for example a sub-part of the whole of the set of integers). In this case, we will focus on the succession or the precedence of the trace observed (there is no chronological time).

In the current uses of the traces for the CEHL, collected situations are contrasted: from "we take what we have in well specified formats, what is called the logs" to "we scrupulously instrument the environment to recover the observed controlled and useful for different actors (learner and tutor). The first step consists of modeling the raw data contained in the log file. It is necessary to be able to collect the traces files containing at least, the following elements: time for the start date of the action, codes action which consists in codifying the learner's actions and learner concerned.

In our system IDCBR-MAS a target case or target trace is represented by a trace learner in progress when interacting with the Moodle platform. The cases sources are previous traces learners that are stored in database. The cases sources are traces left by the learners which followed the same training on the Moodle platform. The following figure shows the target trace structure:

```

resse H3zouhair copie ecran MAS-IDCBR\xml\user9.xml
<?xml version="1.0" encoding="utf-8" ?>
<logs>
- <user>
  <id>5</id>
  <nom>zouhair abdelhamid</nom>
</user>
+ <log>
- <log>
  <id>550</id>
  <course>JADE</course>
  <date>lun 30/9/2013 0:3</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>

  <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scm/content/1/les_protocoles_fipaquery
  <date_second>1380492184</date_second>
  <duree_second>1380492550</duree_second>
</log>
- <log>
  <id>551</id>
  <course>JADE</course>
  <date>lun 30/9/2013 0:9</date>
  <action>launch</action>
  <ip>127.0.0.1</ip>
  <info>http://127.0.0.1/moodle/pluginfile.php/25/mod_scm/content/1/la_norme_fipa.html</info>
  <date_second>1380492558</date_second>
  <duree_second>1380492758</duree_second>
</log>

```

Fig. 4. Learner traces when interacting with the platform Moodle

The trace can be written as follows: ((OA,1, (Q A,1,1, VA,1,1),..., (QA,1,d, VA,1,d), (OA,2, (QA,2,1, VA,2,1),..., (QA,2,d, VA,2,d)),..., (OA,n, (QA,n,1, VA,n,1),..., (QA,n,d, VA,n,d)))

We developed a module in Moodle platform that can be the interface between the Moodle server and our IDCBR-MAS system. This module includes an xml file, which contains traces left by all learners in the Moodle log file and also contains the datalib file. The module uses the same Moodle database. The datalib file of Moodle platform has been modified in order to be able to record and save all traces of learners connected to the Moodle platform. The following figure shows the datalib file.

```
function insert_xml($log)
{
    global $DB, $CFG;
    if($log['userid']!=1 && $log['userid']!=0 && $log['userid']!=2)
    {
        $nomfichier=$CFG->dirroot."/xml/users.xml";
        if (file_exists($nomfichier))
        {
            $userid=$log['userid'];
            $nomfich=$CFG->dirroot."/xml/user".$log['userid'].".xml";
            $sql_request = "SELECT u.* $ctxselect FROM (user) u WHERE u.id=".$log['userid'];
            $result = $DB->get_records_sql($sql_request, null, null);
            if($result != null)
            foreach($result as $us)
            {
                $user=$us->firstname." ".$us->lastname;

                $sql_request = "SELECT MAX(l.id) as max $ctxselect FROM (log) l WHERE l.userid=".$log['userid'];
                $result3 = $DB->get_records_sql($sql_request, null, null);
                if($result3 != null)
                {
                    foreach($result3 as $l)
                    {
                        $id=$l->max;
                        $sql_request = "SELECT c.* $ctxselect FROM (course) c WHERE c.id=".$log['course'];
                        $result2 = $DB->get_records_sql($sql_request, null, null);
                        if($result2 != null)
                        foreach($result2 as $c)
                        {
                            $course=$c->fullname;
                        }
                    }
                }
            }
        }
    }
}
```

Fig. 5. datalib file, version IDCBR-MAS

In the next section, we present our Model based on AUML methodology. AUML or agent UML is a support notation for agent-oriented Multi-agents systems development. It consists in using the UML modeling language and extending it in order to represent agents, their behavior and interactions among them.

V. IDCBR-MAS SYSTEM MODELING

Our system IDCBR-MAS is composed of multiple interacting intelligent agents; it supports the specification, analysis, design and validation of our systems. We present the sequence diagram of the various interactions carried out between the various actors of the platform.

A. Presentation of the situation:

The presentation of the situation (learner's traces) by the platform is a task managed by several agents of the presentation layer of our system IDCBR-MAS. These agents are responsible for the update of the traces. The following sequence diagram illustrates the process of the situation presentation of the learner's traces.

Firstly the Request Agent addresses a request to server in order to retrieve the learner's traces left by the learner during the learning session and sending it to the Generator Agent, this last created/update the Traces Agents-L1: Two cases of figure are presented during the checking, if the Traces Agents-L1 (i) related to the learner i exists then the Traces Agents-L1 (i) will be updated, else the Generator Agent create a new Traces

Agents-L1 (i) able to represent the learner i. the process will be re-run each time there is a change in the learner's traces.

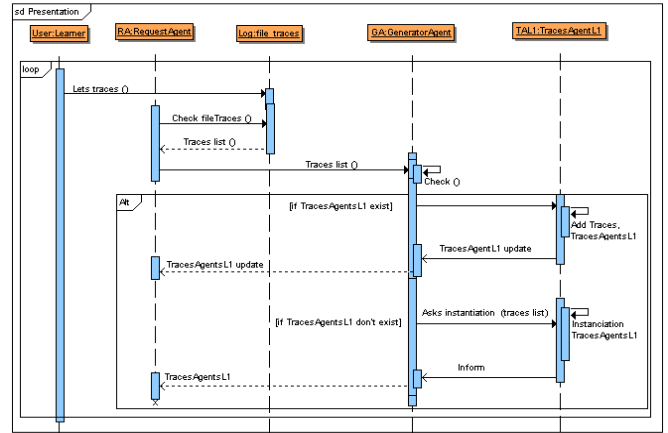


Fig. 6. The sequence diagram of the case presentation in IDCBR-MAS.

B. Interpretation of the situation

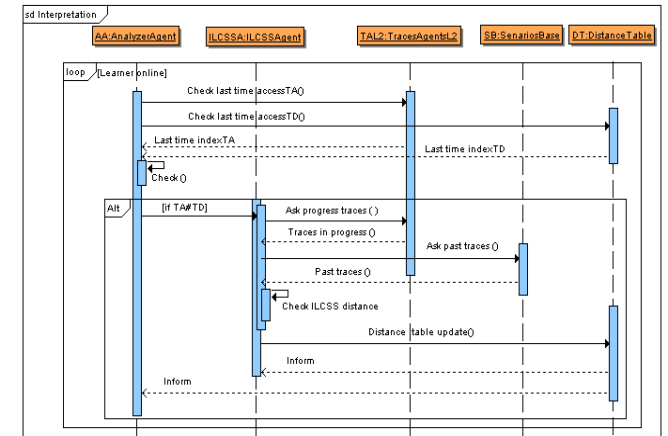


Fig. 7. The sequence diagram of the Interpretation case in IDCBR-MAS

Firstly the Analyzer Agent (AA) addresses a request to the Traces Agents-L2 and to the Distance Table in order to retrieve two chronological dates TA: the last update date in the traces file and DT: the last update date of the Distance Table. The Analyzer Agent check If TA= DT. If the two dates are not equal then the Analyzer Agent ask the ILCSS Agent to update the distance table which contains the distance between the current situation Traces Agents-L2 and the scenario stored in memory. This is based on the similarity measures ILCSS. The agent also asks periodically the Request Agent if there is any change in the learner's traces, whether the process will be re-executed.

First of all the Reuse Agent ask the Traces Agents-L3 to retrieve the current traces with the associated scenarios (the associated scenarios to the current traces are the scenarios that are very similar at learner's traces or target, based on the similarity measures ILCSS). Then the Evaluate Agent checks the Distance Table. If necessary the Reuse Agent asks the ILCSS Agent asks to check and update all distances between the current situation and scenarios stored in memory.

C. Prediction of the situation

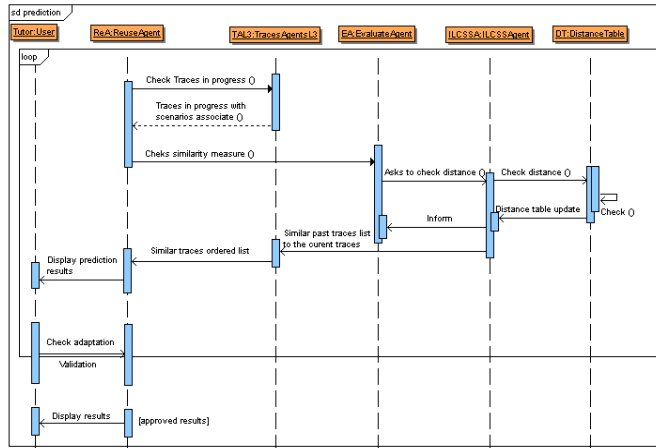


Fig. 8. The sequence diagram of the case prediction in IDCBR-MAS system

VI. IDCBR-MAS SYSTEM DEVELOPING

We developed our framework IDCBR-MAS based the JADE Agent Platform (Java Agent DDevelopment Framework). For the development of interfaces, we chose the languages Java, PHP and the tools EasyPHP, Apache, MySQL, phpMyAdmin.

A. Inter-Agent Communication in IDCBR-MAS

In order to supervise and control the communication and the IDCBR-MAS agents' behavior, we use Remote Monitoring Agent (RMA) of JADE platform. RMA is a graphical console for platform management and control. The RMA console is able to start other JADE tools. It a monitoring and debugging tool, made of a graphical user. It is able to displays the flow of interactions between agents in our IDCBR-MAS platform. The following figure shows the interactions between IDCBR-MAS agents.

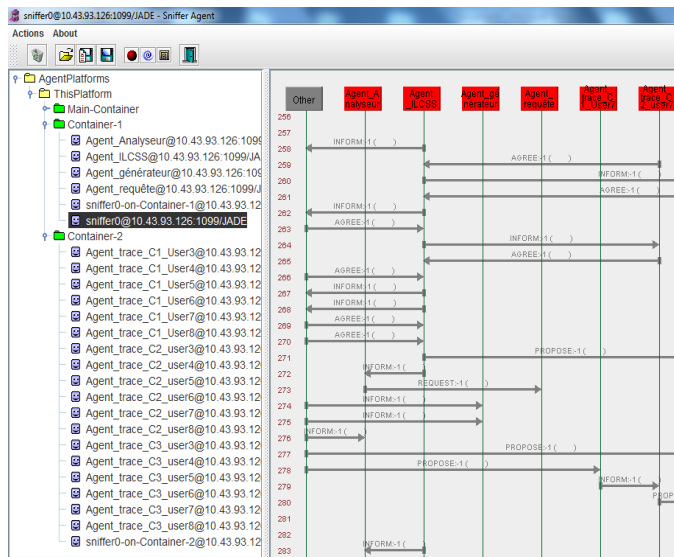


Fig. 9. Inter-Agent Communication in IDCBR-MAS

B. Monitoring the activity and communication between agents in IDCBR-MAS

This tool makes it possible to monitor the life cycle and communication of our agents: Sending and Receiving Messages by these agents. It is also possible to display the list of all the messages sent or received, completed with timestamp information in order to allow agent conversation recording and rehearsal. For example, the following figure shows the state as well as the transmitted/received messages for the ILCSS Agent of our IDCBR-MAS framework.

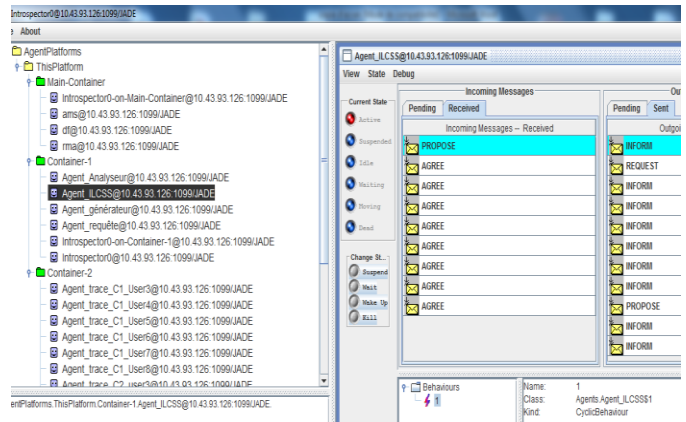


Fig. 10. Transmitted/received messages for ILCSS Agent

C. Distance between the target and previous traces

After the registration of a learner on the IDCBR-MAS platform, the learner will be able to run Moodle from our platform and subsequently launch a learning session. The tutor follows progressively the training of the learner.

Target user	Previous users	Update target case score	Retrieve score	Adaptation score	Distance between target case and previous case
Oussama Oussama	Mimoun zakhenini	44	23	8	0.757
Wahid abdelhamid	Hasani Ali	44	23	8	0.453
Rachid Amernis	Zouhair abdelhamid	44	23	8	0.343
saïd zouhair	Hasani Ali	44	23	8	0.476
Abdelaziz Zitan	Rachid Amernis	44	23	8	0.765
Walid Allouch	Saïd zouhair	44	23	8	0.124
Youssef Kassi	Abdelaziz Zitan	44	23	8	0.876
Hajar talab	Talibi Hamza	44	23	8	0.945
Youssef Kassi	Youssef Kassi	44	23	8	0.344
Hajar talab	Hajar talab	44	23	8	0.788
Rachid Hadj	Rachid Hadj	44	23	8	0.324

Fig. 11. Distance between the target and previous traces

All interactions, actions and productions of the learner are recorded on the log file in the Moodle database. Our system retrieves these traces through agents' interfaces permanently, and then they will be treated by the platform. In the figure we have a target case (traces left by target learner) and we have previous traces (traces left by previous learners). The update target case score present the number of update in the target case; the retrieve score present the number of re-retrieve of the previous cases very similar to target case by the agents of IDCBR-MAS platform. The distance between the target trace and past traces are calculated by the ILCSS Agent. These distances will be used as a key element in predicting of the

situation achieved by the adaptation agent. The system proposes to the tutor a list of the similar traces to the target trace in order to choose the best similar traces.

D. Distances curves between the target and previous traces

The following figure displays the distances curves between the target and previous traces in order to show the distance between them, these curves are generated in real times starting from the results of retrieval phase. These curves display also the history of these distances. For Tutor, the distances curves present very important information about the change of the distances database. The Tutor will be able to take her decision and to choose the trace most similar to the target trace.

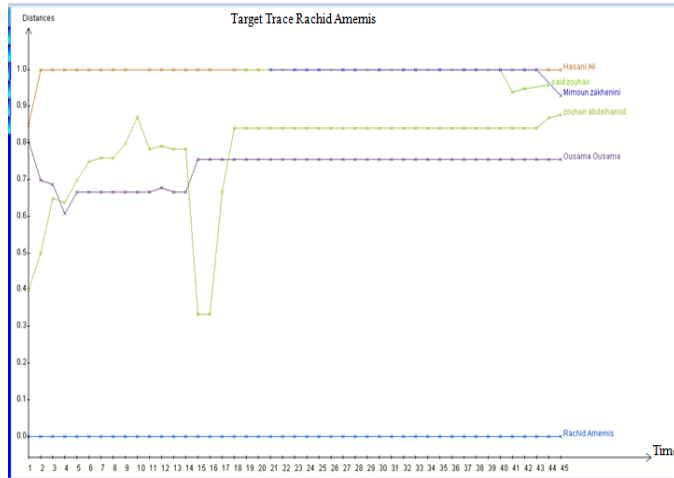


Fig. 12. Distances curves between the target traces and previous traces

VII. IDCBR-MAS & OTHER CBR SYSTEMS

Several researchers have focused on classical versus dynamic CBR architectures where target case are static versus dynamic, but all these systems have been used static CBR cycle. Consequently, the Incremental Dynamic CBR approach has been proposed as an appropriate alternative, which have demonstrated its efficacy. For example, In our approach the evaluation of the similarity between the target case and similar past cases is a process continues and the retrieve step of the CBR cycle take into account the change in the target case in real-time. Finally our system is founded on a dynamic retrieving method. The following table (Table II) shows a summary of the CBR systems.

TABLE II.
CBR SYSTEMS CLASSIFICATION

Target case	CBR Cycle	Classical CBR Systems	CBR-MAS	CBR-Agent
Static	Static	CHEF[13], CREEK [3], CASEY [15], RADIX [8]	CCBR[21], AMAL [27]	ProCLAIM [29]
Dynamic	Static	REBECAS [18], AuRA [16] , SAPED[2], CASEP2 [33], SBR[4]	CICLMAN [28], RoBoCats [20] , S-MAS [24]	MCBR [17], CBR-TEAM[26]
Dynamic	Dynamic			IDCBR-MAS

VIII. CONCLUSION AND FUTURE WORK

Our system allows connecting and comparing the current situation (target trace) to past situation (previous traces) that are stored in a database. The continuous analysis of information coming from the environment (learner's traces) makes it possible to suggest to various actors (learners and tutor) possible evolutions of the current situation.

The Multi-Agents architecture that we propose is based on three layers of agents with a pyramidal relation. The lower layer allows building a representation of the target case. The second layer implements a dynamic process: search for past situations similar to the current one. Finally, the prediction layer captures the responses sent by the second layer to transform them into actions proposed either by virtual tutor, or/and human tutor.

We have presented systems founded of Incremental and Dynamic Case Based Reasoning and we have also clarified that the CBR-based applications can be classified according to the study area: CBR for static situations and CBR for dynamic situations. In our situation, we have used a Dynamic system IDCBR-MAS, with a dynamic CBR cycle in order to push the limits of CBR cycle static. In fact, the current situation (target case) is a trace that evolves; the case based reasoning must take into account this evolution incrementally. In other words, it shouldn't consider each evolution of the trace as a new target case.

Our future work follows two different ways. First, we would like to use our framework in real experiment with e-learning platform of our university. Secondly, in the second part of our perspective, we will try to implement our approach in the field of Geographic Information Systems (GIS).

REFERENCES

- [1] A. Aamodt and E. Plaza, "Case-Based Reasoning : Foundational Issues, Methodological Variations, and System Approaches", AICom-Artificial Intelligence Communications, IOS Press, Vol. 7: 1, pp. 39-59, 1994.
- [2] A. Aich, Reconnaissance et prdiction de situations dynamiques : application l'assistance de personnes handicapées moteur, Thse de doctorat, Universit de Technologie de Troyes, 2007.
- [3] A. Aamodt, Knowledge-Intensive Case-Based Reasoning and Sustained Learning. Proc. of the 9th European Conference on Artificial Intelligence, ECCBR04, Lecture Notes in Artificial Intelligence, Springer, 2004.
- [4] C. Baptiste, Vers la gestion de scnarios pour la reconnaissance et l'anticipation de situations dynamiques , Universit de Technologie de Troyes, France, 2011.
- [5] P. Brusilovski, "Distributed Intelligent Tutoring on the Web". 8th World Conference of Artificial Intelligence in Education. IOS Press. pp.482-489, 1997.
- [6] A. Cordier, B. Mascret, A. Mille, Dynamic Case Based Reasoning for Contextual Reuse of Experience, Case-Based Reasoning Workshop, ICCBR 2010, Cindy Marling ed. Alessan-dria, Italy. pp. 69-78. 2010.
- [7] G. Constantino, D. Suthers and J-I. Icaza, Designing and Evaluating a Collaboration Coach: Knowledge and Reasoning, Proceedings of the Artificial Intelligence in Education, AI-ED, J.D. Moore et al (Eds). The Netherlands: IOS Press. pp. 176-187, 2001.
- [8] F. Corvaisier, Mille A., Pinon J.M., Radix 2, assistance la recherche d'information documentaire sur le web, In IC'98, Ingnieur des Connaissances, Pont--Mousson, France, INRIALORIA, Nancy, 1998.
- [9] M. Ennaji, H. Boukachour, P. Grav, Une architecture Multi-Agent pour la pedagogie de la formation distance, In: MOSIM 2006, Rabat, Maroc.

- [10] E. M. En-Naimi, A. Zouhair, B. Amami, H. Boukachour, P. Person and C. Bertelle, "Intelligent Tutoring Systems Based on the Multi-Agent Systems (ITS-MAS): The Dynamic and Incremental Case Based Reasoning (DICBR) Paradigm and the Inverse Longest Common Sub-Sequence (ILCSS) in the CEHL", *The IJCSI*, Vol. 9, Issue 6, No 3, pp. 112-121, November 2012.
- [11] C. Frasson, L. Martin, G. Gouarderes and E. Aïmeur, "A distance learning Architecture Based on Networked Cognitive Agents", 4ème International Conference on Intelligent Tutoring Systems- ITS'98, San Antonio, USA, 1998, *Lecture Notes in Computer Sciences* 1452, Springer Verlag, pp. 142-151.
- [12] B. Fuchs, J. Lieber, A. Mille and A. Napoli, "Une première formalisation de la phase d'élaboration du raisonnement à partir de cas", Actes du 14ième atelier du raisonnement à partir de cas, Besançon, mars, 2006. [see url : <http://www.lab.cnrs.fr/RaPC2006/programme.html>].
- [13] K.J. Hammond, CHEF: a model of case-based planning, *Proc. Of AAAI86*, Morgan Kaufman, pp. 267-271, 1986.
- [14] J. Kolodner, , Case-Based Reasoning, Morgan Kaufmann, San Mateo, UCA, 1993.
- [15] P. Koton, Using experience in learning and problem solving, PhD Thesis, Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge Massachusetts, USA 1988.
- [16] M. Likhachev, M. Kaess, and R.C. Arkin, Learning behavioral parameterization using spatiotemporal case-based reasoning, The IEEE international conference on robotics and automation (ICRA02), Washington, D.C, 2002.
- [17] D. Leake and R. Sooriamurthi, When two case bases are better than one: Exploiting multiple case bases, In *ICCB*, pp. 321-335, 2001.
- [18] S. Loriette-Rougegréz S., "Raisonnement à partir de cas pour les évolutions spatiotemporelles de processus", *Revue internationale de géomatique* , vol. 8, n° 1-2, 1998, p. 207-227.
- [19] J-C. Marty and A. Mille, , Analyse de traces et personnalisations des environnements informatiques pour l'apprentissage humain, Edition Lavoisier, 2009.
- [20] C. Marling, M. Tomko, M. Gillen, D. Alexander, D. Chelberg, Case-based reasoning for planning and world modeling in the RoboCup small size league, in: *Workshop on Issues in Designing Physical Agents for Dynamic Real-Time Environments: World Modeling, Planning, Learning, and Communicating (IJCAI)*, 2003.
- [21] L. McGinty and B. Smyth, Collaborative case-based reasoning: Applications in personalized route planning. In *ICCB*, pp. 362-376, 2001.
- [22] A. Mille, , From case-based reasoning to traces-based reasoning, *Annual Reviews in Control*30(2):223-232, ELSEVIER, ISSN 1367-5788. 2006.
- [23] J. Piaget, *Psychologie et pédagogie*, Paris: Denol-Gonthier, 1969.
- [24] I. Pinzón Cristian, B. Javier, F. Juan, M. Juan, B. Corchado, S-MAS: An adaptive hierarchical distributed multi-agent architecture for blocking malicious SOAP messages within Web Services environments, *Expert Systems with Applications* Volume 38, Issue 5, May 2011, pp. 5486-5499.
- [25] E. Plaza and L. McGinty, , Distributed case-based reasoning, *The Knowledge Engineering, Review*, Vol. 00:0, 14.c 2005, Cambridge University Press, 2005.
- [26] M. Prasad, V. Lesser, and S. Lander, Retrieval and reasoning in distributed case bases. Technical report, UMass Computer Science Department, 1995.
- [27] S. Ontañón and E. Plaza, Learning and Joint Deliberation through Argumentation in Multi-Agents Systems, in *AAMAS'07 Proceedings of the 6th international joint conference on Autonomous agents and Multi-Agents systems*, Honolulu, HI, USA, May 14-18, 2007.
- [28] L. k. Soh and J. Luo, Combining individual and cooperative learning for multiagent negotiations, In *Int. Conf. Autonomous Agents and Multiagent Systems AAMAS03*, 2003.
- [29] P. Tolchinsky, S. Modgil, U. Cortes, and M. Sanchez-marre, Cbr and argument schemes for collaborative decision making. In *Conference on computational models of argument, COMMA-06* (Vol. 144, pp. 71-82), 2006.
- [30] L-S., Vygotski, *Mind in society: the development of higher psychological processes*, Harvard University Press, Cambridge, MA, 1978.
- [31] M. Vlachos, K. Kollios, and G. Gunopulos, Discovery similar multidimensional trajectories. The 18th International Conference on Data Engineering (ICDE02), pages 673-684, San Jose, CA, 2002.
- [32] C. Webber and S. Pesty, , Emergence de diagnostic par formation de coalitions - Application au diagnostic des conceptions d'un apprenant, *Journées Francophones pour l'Intelligence Artificielle Distribuée et les Systèmes Multi-Agents*, Hermes, Lille, pp.45-57, 2002.
- [33] F. Zehraoui, Systèmes d'apprentissage connexionnistes et raisonnement partir de cas pour la classification et le classement de séquences, Thèse de doctorat, Université Paris13, 2004.
- [34] ZOUHAIR Abdelhamid, El Mokhtar EN-NAIMI, Benaïssa AMAMI, Hadhoum BOUKACHOUR, Patrick PERSON, Cyrille BERTELLE, *Intelligent Tutoring Systems Founded on The Multi-Agent Incremental Dynamic Case Based Reasoning, IEEE CIST'12*, Fez, Morocco, 2012.

Abdelhamid ZOUHAIR is a PhD student in Cotutelle between the Laboratory LIST, FST of Tangier, Morocco and the Laboratory LITIS, the University of Le Havre, France, since September 2009.

El Mokhtar EN-NAIMI is a Professor in Faculty of Sciences and Technologies of Tangier, Department of Computer Science. He is a member of the Laboratory LIST (Laboratoire d'Informatique, Systèmes et Télécommunications), the University of Abdelmalek Essaâdi, FST of Tangier, Morocco. In addition, he is an associate member of the ISCN - Institute of Complex Systems in Normandy, the University of Le Havre, France.

Benaïssa AMAMI is a Professor in Faculty of Sciences and Technologies of Tangier. He was an Ex-Director of the Laboratory LIST (Laboratoire d'Informatique, Systèmes et Télécommunications), the University of Abdelmalek Essaâdi, FST of Tangier, Morocco.

Hadhoum BOUKACHOUR and **Patrick PERSON** are Professors in the University of Le Havre, France. They are members in the Laboratory LITIS (Laboratoire d'Informatique, de Traitement de l'Information et Système), The University of le Havre, France.

Cyrille BERTELLE: Professor in Computer Science, Complex Systems Modelling and Simulation. Member of the Research Laboratory LITIS at the University of Le Havre, Normandy, France (RI2C Team). Co-founder of ISCN - Institute of Complex Systems in Normandy. Vice-President of Research and Development at University of Le Havre, France