



Resolution of difficulties approach for a community of practice members: design, implementation and experiment

Rachid Belmeskine^{a,b}, Ahlame Begdouri^a, Dominique Groux-Leclét^b

^aSIA Laboratory. Faculty of Science and Technology, Fez, Morocco

^bMIS Laboratory. University of Picardie Jules Verne, Amiens, France

ABSTRACT

MAETIC is a pedagogical method intended to implement the pedagogy of a group per project. When a teacher deploys this method in his classroom, he may encounter difficulties due to the lack of experience and may need help. To support the community of practice of teachers that use the MAETIC method, we design a Community Computing Environment (CCE). This CCE consists of a kernel, a RD (Resolving Difficulties) layer and two adaptation's layers. In this paper we detail the kernel that incorporates the basic features of a community environment and the RD layer that permits to assist teachers in solving their problems. This layer is based on Case-Based Reasoning (CBR).

Indexing terms/Keywords

Case-Based Reasoning, Communities of Practice, Teacher training, Project based pedagogy, Knowledge capitalization.



Council for Innovative Research

Peer Review Research Publishing System

International Journal of Research in Education Methodology

Vol.4, No.1

editor@ijrem.com

www.cirworld.com, www.ijrem.com

1. INTRODUCTION

The research presented in this paper concerns the development of a Community Computing Environment (CCE) based on the social web 2.0 technologies. This environment supports a Community of Practice (CoP) [4][6][8][19][23].

This community consists of teachers users of a teaching method called MAETIC (French acronym: Méthode pédAgogique instrumEntée par les Technologies de l'Information et de la Communication). MAETIC is intended to implement the pedagogy of a group per project. The experiments have shown that when a teacher deploys this method in his classroom, he may encounter difficulties due to the lack of expertise and experience and may need help. More experienced teachers can provide him this help. The figure 1 shows an example of a difficulty encountered by a teacher:

Which must by my attitude during the meeting: do I let make and do I come when me en called or I systematically make a point with each group ?
My class consisted of 30 students divided into 5 groups and my teaching domain is computer science.

Fig 1: An example of a difficulty encountered by a teacher

In this context, we would like to offer for this community a CCE to enable them to share their practices and make requests for assistance in resolving difficulties. The structure of our environment is composed by four layers: the kernel layer, the RD layer (Resolving Difficulties), the AI layer (Adaptation of the Interactions) and the AP layer (Adaptation of the Presentation of the HMI).

The kernel layer includes the basic features of a community environment. The RD layer is based on Case-Based Reasoning (CBR) to help members to solve their difficulties. The AI layer supports the adaptation of members' interactions to their contexts of use. Finally, the AP layer provides an interface layout suited to classical and mobile terminals.

The objective of this paper is to describe the kernel of the CCE and the RD layer. Thus, we first present the context of our research, after that we present the kernel and the RD layer, and then we focus on the retrieve phase of the CBR. Finally, we present the experimental validation of the RD layer.

2. RESEARCH CONTEXT

2.1. The community of practice

The Community of Practice (CoP) is a concept derived initially from the research of Lave and Wenger [11]. It is a form of gathering of professionals interested in the same practices. Wenger defined it as a group of people who share a concern or a passion for something they do and learn how to do it better as they interact regularly [22].

Three characteristics of the CoP are considered as crucial [20]:

- The domain – joint enterprise: Members of a community of practice have a shared competence and shared domain of interest that distinguishes them from other people.
- The community – mutual engagement: Relationships between the members are guided by the learning aspect of interaction. Sharing knowledge and information, engage in joint activities and discussions, and helping each other.
- The practice – shared repertoire: Members of a community of practice are practitioners. They develop a shared repertoire of resources: experiences, stories, tools, ways of addressing recurring problems, in short a shared practice. This takes time and sustained interaction.

Lave and Winger [11] show that the interaction, the inter-individual relationship and cooperation result in the production of tacit knowledge, difficult to formalize, to store and to recover later.

The needs of CoPs can be expressed differently, according to several factors, among these factors: the context of emergence, the history, the number of members, but the most important is the domain of professional's practices. The objectives of the CoPs are changed according to their needs but the CoPs have always a main common objective which is the establishment of a structure where tacit and explicit knowledge are shared and exchanged among various members within a given domain.

At first, the concept of the CoP was not related to ICTs (Information and communications technologies). But the progresses of the ICTs have given new opportunities to develop the CoPs. Indeed, the ICTs allow to transcend the barriers of time and space, and to group individuals according to their interests, not their geographic location.

In this context, communities have used a variety of communication modes, from traditional media such as telephone, fax, or video conferencing, to more sophisticated technological tools such as email, Internet, intranet, or discussion forums [21]. With the advent of Web 2.0 tools, such tools as wikis, weblogs (blogs), forums and social networking sites have become the preferred tools of several existing CoPs and have given birth to new CoPs.

These web 2.0 tools have several advantages, they have free versions, they often require minimal configuration. They also have limitations, the main ones are that these tools do not allow sophisticated personalization and are not always good at providing easy navigation for large amounts of resources. These limitations gave birth to several researches related to the personalized computing environments for CoPs. Among these researches, we find the PALETTE project (<http://palette.ercim.org/>, 2006) which aimed at equipping CoPs in the domains of management, engineering and education, with a set of tools necessary to their operation. A. Chikh and L. Berkani [5] proposed a computing environment to a CoP of the actors of an online learning system. In other research, R. W. Helms and al. [9] proposed a portal dedicated to a CoP of employees in order to improve knowledge sharing in an offshoring context.

Our research is in this direction of the personalized computing environments for CoPs. Indeed, we interest to develop a personalized computing environment for a CoP of teachers that use a pedagogical method called MAETIC.

2.2. The CCE-MAETIC

MAETIC (French acronym: Méthode Pédagogique Instrumentée par les TIC) is a pedagogical method which was developed through collaboration between researchers from the UPJV (Picardie Jules Verne University) and the ULCO (Littoral Opal Coast university) [18]. This method target students of professional trainings and aims, for them, the acquisition of professional know-how by using an approach based on group project pedagogy.

To use MAETIC, a teacher develops a pedagogical device, which is deployed for his students in the classroom. In 2008, experiments have shown that teachers needed support throughout the deployment phase [13]. They specifically want to benefit from an assistance to solve their difficulties. The experiments have also shown that such aid could be provided by more experienced teachers in the use of the method.

These teachers, users of the MAETIC method, are the CoP-MAETIC members targeted by our research. Thus, these teachers are considered as learners in the CoP MAETIC. In this way our research is focused on supporting the CoP MAETIC by the design of a Community Computing Environment (CCE) that uses Web 2.0 technologies and that is adapted to the specificities of the CoP MAETIC. We call this CCE: The CCE-MAETIC [2][3]. The main objective of the CCE-MAETIC is to allow the teachers of the CoP MAETIC:

- To exchange their practices about MAETIC.
- To make requests for assistance in resolving encountered difficulties.

In this context, a previous research work has established a classification of identified difficulties categories for MAETIC method. It was elaborated during a gathering of expertise conducted in 2010 by C. Quénu-Joiron and D. Lecllet [13]. They have particularly identified two main classes of difficulties: **the transverse difficulties** and **the difficulties specifically related to a MAETIC method step**. As example, the Figure 2 below shows the transverse difficulties:

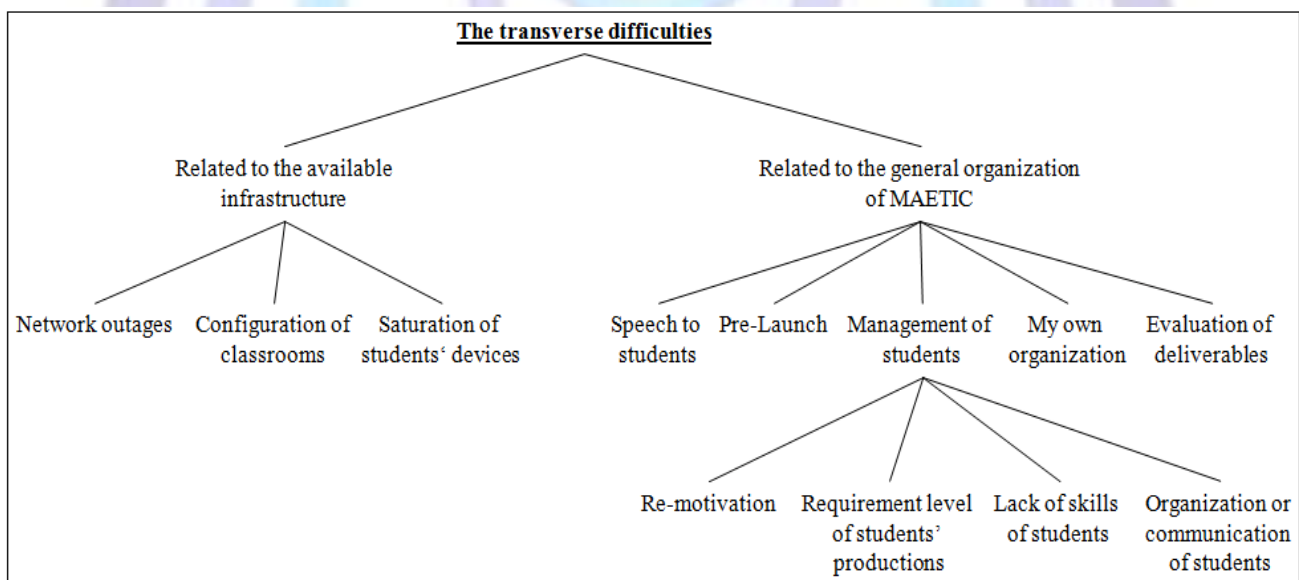


Fig 2: Classification of the transverse difficulties

This classification of the MAETIC difficulties will be exploited in the design of the CCE-MAETIC that we present in the next paragraph.

3. DESIGN OF THE CCE-MAETIC

3.1. Structure of the CCE-MAETIC

The Figure 3 below shows the structure of the CCE-MAETIC.

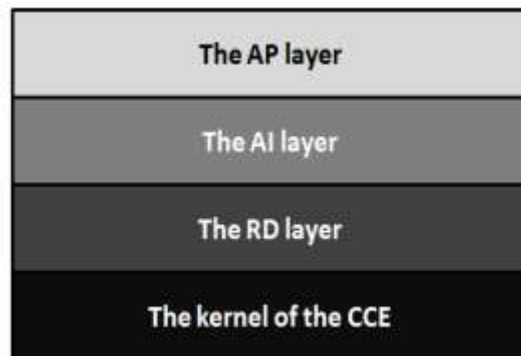


Fig 3: The structure of the CCE-MAETIC

It consists of:

- The kernel that incorporates the basic features of a community environment. This kernel allows the management of the community members, of the links between the members and the management of their conversations.
- The RD Layer (Resolving Difficulties) aims to minimize the number of aid requests on the community space by helping the teacher to solve his difficulties. We rely on difficulties that have already been solved in the past. The scenario of operation of this layer is based on the Case-Based Reasoning (CBR).
- Two adaptations layers: the AI layer and the AP layer. These 2 layers are out of the scope of this paper and concern adaptation aspects of the interactions generated during the exchange between the CoP members.

3.2. The kernel

The kernel of the CCE-MAETIC offers the basic functionalities of a community environment adapted to the specificities of the CoP MAETIC. Among these specificities:

- Only the teachers recognized as users of the MAETIC method can access.
- Each member is connected with the others because the community is small.

The kernel is essentially based on the publication of contents. A content can be an article or a Conversation Thread (CT). A CT is initiated by the publication of a difficulty by a member, and then it is continued through comments posted by other members. Each member has a personal page, which contains his publications. A content can then be commented by all members of the community. Each content posted by a member on his personal page is visible to all members of the community through an Actualities Thread (AT). The kernel doesn't allow a member to post a request for help directly on the personal page of another member.

The main objective of the kernel is to facilitate the interactions between teachers members of the CoP in order to find solutions for their difficulties and to capitalize their practices related to MAETIC. It allows managing 3 types of users:

- The user "Member": who is a teacher that uses the MAETIC method.
- The user "Expert": who is a teacher "expert" in the use of MAETIC. He has the right to validate the proposed solutions to the difficulties faced by the teachers of the CoP.
- The "Administrator": the one who manages the EIC-MAETIC and ensures its proper functioning.

Figure 4 below presents the use case diagram of the kernel and shows its functionalities.

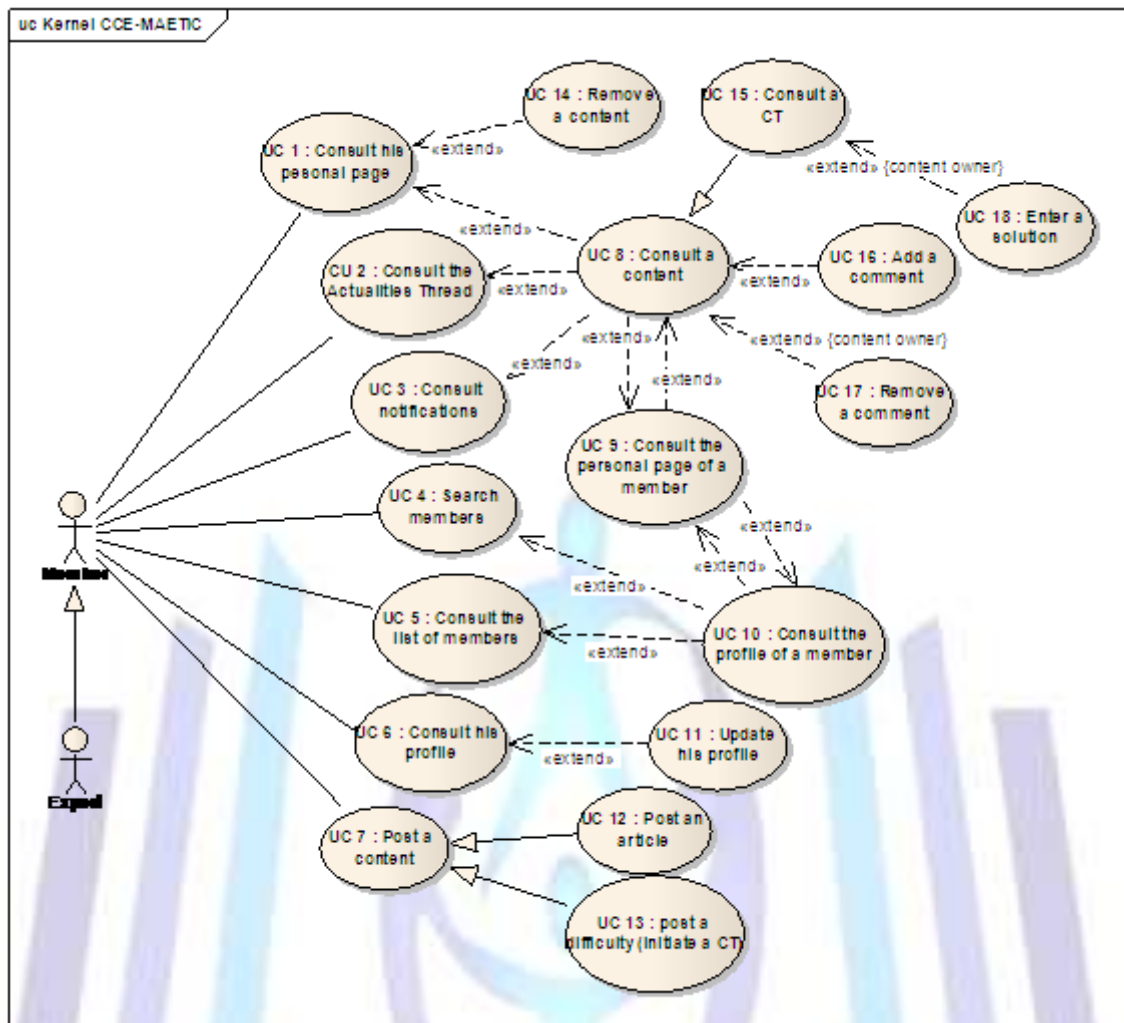


Fig 4: The use case diagram of the kernel

A member can:

- Consult his personal page (UC 1). In his personal page, he can consult a content (UC 8) or remove a content (UC 14).
- Consult the notifications (UC 3) that he receives when a new comment is added in a content that is published or commented by him. Through the consultation of a notification, he can then consult the content which is the source of the notification (UC 8).
- Consult the AT (UC 2), which is a page that contains all the new publications and where he can have access to recently published contents.
- Consult the list of the community members (UC 5), or search for a member (UC 6). Through these 2 UC's, the member can consult the profile of another member (UC 10) and through that he can consult the personal page of this member (UC 9).
- Consult his profile (UC 6) and from here update it (UC 11).
- Post a content (UC 7) which can be an article, or a difficulty to initiate a CT.

In addition:

- When a member consults the personal page of another member of the community (UC 9), he can consult one of the contents posted on this page or consult the profile of this member.
- When a member consults a particular content (UC 8) he can add a comment, or remove it if it is his own content.
- When a member consults his own CT, he has the opportunity to enter a solution (UC 18). We recall that a CT is initiated by the publication of a difficulty.
- An expert has access to all the functionalities offered to a member.

3.3. The RD layer

3.3.1. The CBR

In order to help the CoP MAETIC's teachers to solve their difficulties, the RD layer relies on the Case base reasoning. The CBR is a problem solving approach based on reuse by analogy from past experiences called cases [1]. The set of the experiments forms a case base. The CBR's is based on five steps that are: **the elaborate**, **the retrieve**, **the reuse**, **the revise** and **the retrain** phases. Few CBR systems make the adaptation and revision phases fully automatic. For most systems, human intervention is necessary.

In the literature, the CBR models are grouped into three major families: structural, conversational and textual [10]. In the structural model, the cases are completely structured and are represented by attributes of simple types. The conversational model is based on the interaction between the user and the system (conversation) in order to gradually identify the problem and to select the most appropriate solutions. The textual model consists of solving problems using experiments that are described in textual documents.

The RD layer implements the CBR's different steps (figure 5). **The elaborate phase** consists of building the target case from the information provided by the teacher about the difficulty encountered. **The retrieve phase** consists of extracting the source cases that are similar to the target case called. **The reuse phase**, allows the teacher to design a solution, using the extracted cases or to provoke a discussion with the experts if the extracted cases do not satisfy his needs. At that time, the system calls the kernel layer. **The revise phase** involves experts who evaluate and validate the solution designed by the teacher. **The retrain phase** allows adding the problem and its validated solution into the case base.

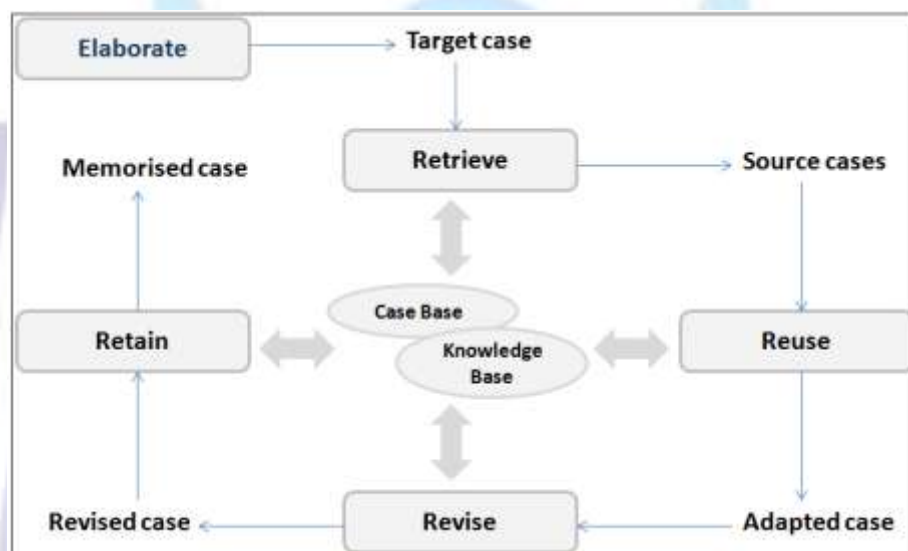


Figure 5. The CBR's cycle

For our RD layer, we use a combination of textual and structural models. Indeed, our retrieve step consists of 2 phases: the first is based on a textual attribute (textual model), and the second is based on simple attributes (structural model). This step is presented in the section 4.

3.3.2. The structure of cases

Generally, a case contains at least two parts: the description of a situation that represents the "problem" and the "solution" used to remedy this situation. Sometimes, the case also describes the consequences resulting from the application of the solution [10].

In our environment, the structure of the cases is based on the structure proposed by C. Quénu-Joiron & D. Lecllet [13] as follows:

- The problem part 'pb' consists of:
 - Context-pb: represents the context of the difficulty and it contains information about the context such as: the teacher's expertise level, the university size, effective, etc.
 - Specif-pb: represents the specification of the problem, and contains the description of the difficulty encountered.
- The solution part 'sol(pb)' consists of :
 - Sol-actions: represents the actions to be performed by the teacher to solve the problem (difficulty encountered).
 - Sol-effet: represents the effects of the solution actions.

- Sol-eval: represents the evaluation of the solution by the experts.

We use the classification of difficulties categories elaborated by C. Quénu-Joiron and D. Leclét [13] for MAETIC method by adding an attribute we called 'TYPE OF DIFFICULTY' in the 'Specif-pb' component. Indeed, this attribute allows to specify the type of the encountered difficulties according to this classification. The use of this attribute allows reducing the number of source cases which we compare with the target case in the retrieve phase. Which positively influences the performance of the RD layer. Figure 6 explains our case structure.

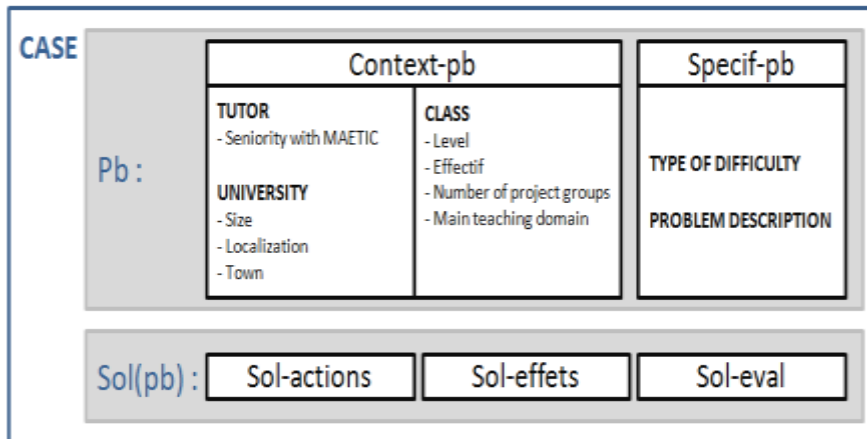


Fig 6. The case structure in the RD layer

3.3.3. Functionalities

The Figure 7 below presents the use case diagram of the RD layer. It shows its functionalities.

A member or an expert can:

- Describe a difficulty and receive as a result similar difficulties to his difficulty (UC 19), this use case is based on the CBR approach that is described above.
- Post his difficulty on the kernel if he is not satisfied.
- Add a solution to the difficulty in case of satisfaction (CU 20). This is made at the reuse step, and then the difficulty will be marked as solved.

An expert can consult the list of resolved difficulties (UC 21) and validate (UC 22) or remove (UC 23) them. These 3 use cases (UC 21, UC 22, UC 23) correspond to the evaluation step. After the validation, the difficulty and its solution are added automatically to the case base (retain step).

The next section presents the retrieve step of the CBR's cycle.

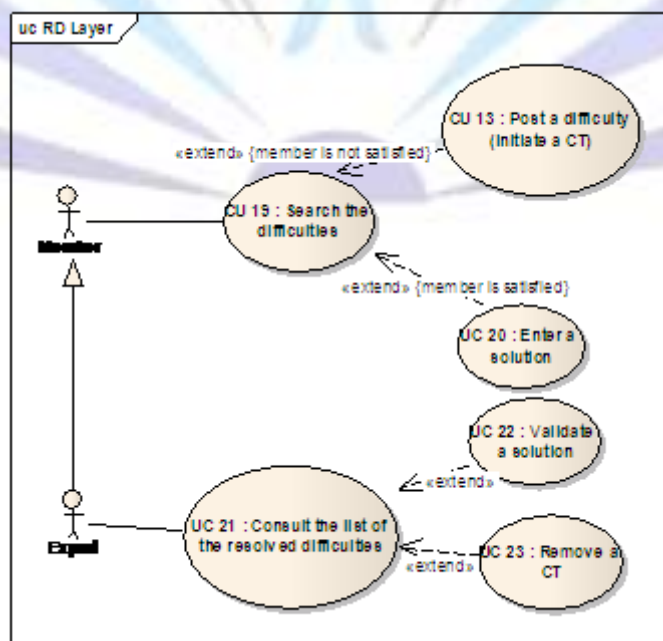


Fig 7. The use case diagram of the RD layer

4. THE RETRIEVE STEP

We remind that the retrieve step of the RD layer consists of extracting the source cases judged similar to the target case. We divide this step into two phases: *research of similar cases*, and *sorting of selected cases*.

4.1. Selection of similar cases

We propose that this phase selects similar cases using only the attributes of 'Specif-pb' (the type of difficulty and its description). The system measures the textual similarity between the 'PROBLEM DESCRIPTION' attribute value of the target case and those of the source cases which have the same type of difficulty 'TYPE OF DIFFICULTY' of the target case. This phase returns the cases considered similar to the target case and their indices of similarity with the target case. The maximum number of cases returned is 10.

We argue our choice to use only the 'specif-pb' attributes for research by the fact that if we use all the attributes of the "pb", it might return source cases that are very similar to the target case in term of context but very different from the difficulty described. For example, if D1 and D2 are two different difficulties encountered in the same context then the two difficulties will have similar values for the attributes 'context-pb' and different values for the 'PROBLEM DESCRIPTION' attribute. In addition, if the similarity in the RD layer is based on all the attributes of 'pb' the two difficulties D1 and D2 will be considered as similar because the difference between the values of the 'PROBLEM DESCRIPTION' attribute of the two corresponding cases will be ignored compared to the similarities between the values of all the other context attributes.

4.2. Textual similarity

In the case of our environment, the description of a difficulty is described in natural language and this description is considered as the attribute most relevant to compare two cases. In this subsection we focus on textual similarities between the difficulties' descriptions.

In order to represent the descriptions by linguistic units, several choices are possible: representation by words, by N-gram [17], and representation by group of words [12]. We eliminated the third one because the descriptions of our difficulties are small which would generate too low frequencies to be exploitable. The N-gram representation was eliminated because of its large execution time compared to the execution time of the word representation we adopt.

Regarding the information retrieval, considering the sizes of the descriptions and if we suppose a certain degree of independence between the words constituting a description then the most suitable model of information retrieval is the vector model [14]. Therefore, our approach for calculating the similarities between the descriptions of the difficulties is based on a representation by words as linguistic units and on the vector model for information retrieval. In the following, we present the two main processes of the calculation of textual similarity that are: the indexation, and the measure of similarity.

The indexation process is decomposed into two sub-processes: the extraction of indexation terms and the vectorization of the descriptions of difficulties.

Before addressing these two processes, we define:

- $C = \{c_1, c_2, \dots, c_k\}$: Our initial base case, where k is the number of cases.
- $D = \{d_1, d_2, \dots, d_k\}$: The set of the textual descriptions of C , so that d_i is the textual description of c_i .

4.2.1. Indexation: Extraction of indexation terms

This sub-process modifies the descriptions in order to facilitate their usage in the second sub-process. It consists of 3 stages:

- Tokenization: consists of dividing the text into linguistic units (words).
- Elimination of the "stop words": consists of removing the terms that are unnecessary to use for indexation or for research. For example: 'a', 'the', 'of', 'who', 'for', 'his'...
- Lemmatization: allows reducing all the forms of a word in one, called lemma. For example: 'walk' is the lemma of: 'walk', 'walked', 'walks' and 'walking'. It permits an important gain in computing time.

This sub-process produces a set of terms that should be completed manually by adding or eliminating certain terms. The final set consists of a set of words called indexation terms. We note this set of terms $T = \{t_1, t_2, \dots, t_n\}$, where n is the number of indexation terms.

4.2.2. Indexation: Vectorization of the descriptions of difficulties

This sub-process seeks a representation of the descriptions that would facilitate their use to measure the similarities between them.

Here, we represent the descriptions of difficulties by v_i vectors in a vector space whose the basis corresponds to the indexation terms retained after the first sub-process. The dimension of this space is the number of indexation terms. For each element of v_i corresponds a weight w_{ji} which represents the weight of the term t_j in d_i . This weight can be represented in several ways: Boolean representation, frequency or TF-IDF representation [15].

The Boolean representation marks the presence or the absence of a word. It does not take into consideration the number of occurrences of a term in a description. Consequently, some words can be judged as more important than their real interest, while others may have a reduced interest.

The frequency representation allows measuring the importance of a word based on its number of occurrences in a description. The TF-IDF representation keeps the same principle of the frequency representation and involves the IDF factor which is used to give greater prominence to words that appear less frequently. Indeed, a term that comes too often in the descriptions of difficulties is not significant. This is why, in our case, we choose to use the TF-IDF representation instead of the frequency representation.

According to this representation, a weight w_{ij} is calculated as follows:

$$w_{ij} = tf_{ji} * idf_j \quad \text{where} \quad (1)$$

- tf_{ji} is the frequency of occurrence of the term t_j in d_i .
- $idf_j = \log(N/n_j)$, where N is the number of d_i (in our case $N=k$), and n_j is the number of the descriptions d_i that contain t_j . idf_j represents the inverse of the frequency of t_j in D .

Figure 8 below summarizes the indexation process.

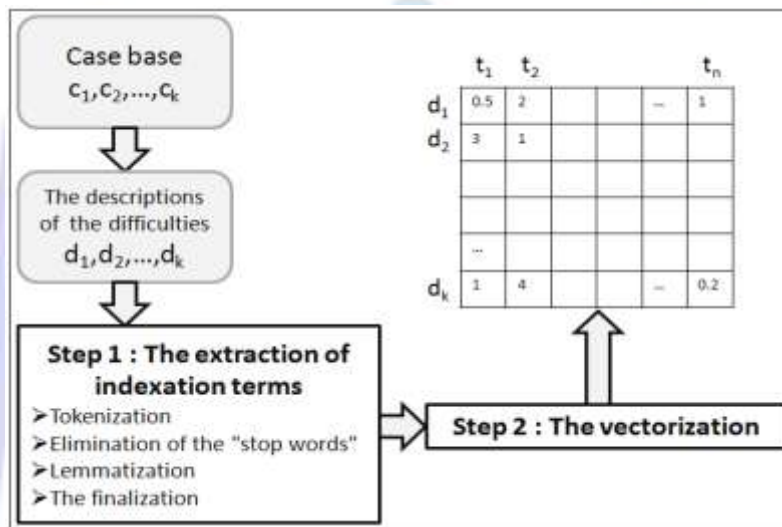


Fig 8. Summary of the indexation process

4.2.3. Similarity measure

To measure the similarity between two descriptions of difficulties, we must calculate the similarity between their associated vectors. Several measures that allow calculating the similarity between two vectors are possible (the Euclidean distance, the chi-square distance, etc.). We chose the cosine measure [16] which, unlike other distance based measures, does not take into consideration the lengths of the vectors. This would mean in our case, that two descriptions of the same difficulty will be judged similar independently of their number of words.

d_x and d_y are two descriptions of difficulties, and v_x and v_y their associated vectors. To calculate the similarity between d_x and d_y , we calculate the cosine between v_x and v_y by the following formula (2):

$$sim(d_x, d_y) = \cos(v_x, v_y) = \frac{v_x \cdot v_y}{\|v_x\| \|v_y\|} \quad (2)$$

More the $sim(d_x, d_y)$ is large, more the descriptions are similar. Contrariwise, more the $sim(d_x, d_y)$ is small, more they are different. We call $sim(d_x, d_y)$ the similarity index between d_x and d_y .

4.2.4. Sorting of selected cases

The second phase in the retrieve step consists of sorting the cases returned (candidate cases) by the first phase (research of similar cases). It consists of sorting these cases using all the attributes of the 'pb' part. These attributes, except the 'PROBLEM DESCRIPTION' attribute, are of simple types which makes it easy to calculate the distance between the values of these attributes. For the 'PROBLEM DESCRIPTION' attribute that is of textual type, we consider the similarity indices calculated in the first phase as distances between the values of this attribute, which brings us to the structural model of CBR.

To calculate the similarity between the target case and a candidate case, we start by calculating the distance between the values of the same attributes. Then, we calculate the overall similarity between these two cases by a weighted sum of the similarity of each attribute. As the attributes are not all equally important, we assign a weight to each attribute of the 'pb' part of the cases based on proposals that will be given by MAETIC experts.

5. VALIDATION AND IMPLEMENTATION OF THE RD LAYER

5.1. Implementation

The RD layer allows a teacher to build a request that describes his difficulty. This layer returns a list of difficulties with their solutions that are judged as satisfactory as possible. The user can choose a solution to his difficulty, or re-post it if he is not satisfied.

This layer is based on the CBR approach cycle. The most important step implemented by this layer is the retrieve step whose the evaluation is the focus in this section.

To implement our system, we choose the use of the Content Management System (CMS): Drupal (<http://drupal.org/>, 2013). We have chosen Drupal because of its power in the implementation of community environments, its active community and its modular architecture that allow evolving the system easily. In addition to that, a set of tools related to the MAETIC method were previously developed with Drupal.

The kernel was developed by using the modules offered by Drupal such as Views, Views UI, Block Search User, Wysiwyg, etc. In order to develop the RD layer, we have decided to implement our module in order to respect the specificities of this layer.

Figure 9 below represents the form used by the module of the RD layer in order to retrieve a target difficulty. This form consists of 4 parts:

- The type part: allows to select the type of the difficulty.
- The description part: allows to enter the title and the description of the difficulty.
- Context information: allows to enter the context information (the fields of this part are not obligatory).

The form is titled 'Form of description of difficulty' and is organized into four main sections:

- Type:** Contains three dropdown menus: 'Type of the difficulty *' (selected: 'Transverse difficulty'), 'Related to the general organization of MAETIC' (selected: 'Pre-Launch of MAETIC'), and 'Pre-Launch of MAETIC'.
- Description:** Contains a text field for 'Title of your difficulty *' (filled with 'Problem with the lectures in the auditorium') and a rich text editor for 'Description of the difficulty *' (filled with 'My module is divided into lectures and TDs, what I can do during the lectures?').
- Information about the class:** Contains four dropdown menus: 'Level', 'Domain' (selected: 'Computer science'), 'Number of students', and 'Number of project groups'.
- Search a solution:** A button at the bottom of the form.

Fig 9. The form of description of difficulty



5.2. Scenario of validation

To evaluate the various operations that must be performed by the system during the retrieve step, we defined the following scenario:

5.2.1. The target difficulty

We use a target difficulty of type 'Pre-Launch'.

We give to the system the following information about our difficulty:

Table 1. Information about the target difficulty

Type	Description	Context information
Pre-Launch	my module is divided into lectures and TDs, what can I do during the lectures?	Domain: computer science

Only one attribute of the context is specified: the domain.

From this information, the system constructs the target case with 'null' value for the non specified context attributes.

5.2.2 The initial case base

We use an initial case base elaborated from a set of difficulties already encountered by teachers that use MAETIC and resolved by experts. This case base consists of 10 difficulties: one case of type 'Network outages', one of type 'Planification step', one of type 'Init step', one of type 'Lack of skills of students', one of type 'My own organization', and 4 cases of type 'Pre-Launch'.

The following table (table 2) lists an extract of these cases without context information because we don't use them during the two first sub-steps of the retrieve step where the selection is based on the type of the difficulty and the textual similarity. We will present these information later for the cases selected in order to use them in the sorting sub-step.

In this table, we put, as example, only one case among the cases which don't have the same 'type of difficulty' than the target case and which will be eliminated during the sub-step of the selection by type.

Table 2. An extract of the initial case base (without context information)

Libellé	Type	
CASE-1	Network outages	Description: Sometimes there are external parameters that hinder the smooth running of sessions: the internet does not work, some websites including blogs are blocked due to wrong setting of filtering software at the server level. Solution: I encountered this problem several times, I recommend the paper logging or the preparation of the work in text format. When the Internet is accessible, they can put their work already prepared on their machine.
CASE-3	Pre-Launch	Description: The number of students is very high for a pedagogy per project: about 60 students. I had to divide them into two groups, I found this a bit tiring. From now I start a new approach: for the group of Tuesday I deploy MAETIC, the other without MAETIC. In order to see the difference between the two methods. Solution: Yes. MAETIC can be supported 10 or 11 groups of 4 students, so the max is 40 students. It is good to divide them into 2 groups. it is Tiring in the beginning, we must be structured, and not let the students dispersed.
CASE-6	Pre-Launch	Description: I could not know in detail the objectives of the courses that are required in the form of scenario. Solution: The objectives of the course are personal to the teacher. Just ask yourself the question "what do I want to convey to my students as knowledge". We have talked about the knowledge areas.
CASE-7	Pre-Launch	Description: I could not predict in advance all the sessions with precision and their activities and how these activities could be adjusted with a preset time. Solution: Step 1 is carried out in the presence of up to 6 hours. Step 2 and 3 are carried out in the presence of up to 6 hours. If students have not finished, they must do so outside the sessions. Step 5 is carried out in the presence of 4 hours.
CASE-10	Pre-Launch	Description: I must do lectures in the auditorium but I do not want to follow during the



		lectures, because the students do not listen to me. Solution: 1. We divide the students into groupes of 4 or 5. 2. We can make them read and resume the PPTs and response to QCMs about what they read. 3. We do the courses about the important concepts.
--	--	--

The choice to use a target difficulty of type 'Pre-Launch' was made in order to have a sufficient number of source cases that have the same 'type of difficulty' than this target difficulty, and then ensure a good evaluation of the sub-step of the retrieve step which is based on the textual similarities.

5.3. The selection sub-step

In the retrieve step, the system starts by eliminating the difficulties that do not have the same type as the target case. Therefore, it keeps only the case: CASE-3, CASE-6, CASE-7 and CASE-10. Then, the system measures the textual similarity between the description of the target case and those of the cases: CASE-3, CASE-6, CASE-7 and CASE-10. The table 3 below shows the similarity indices:

Table 3. The similarity indices

Description of the case	CASE-3	CASE-6	CASE-7	CASE-10
Similarity index with the target case	0	0.204	0	0.34

These similarity indices show that the nearest case to the target difficulty is the CASE-10, followed by CASE-6. Then, the system eliminates the cases that have the textual descriptions which have similarity indices equal to 0 with the description of the target case. It keeps only the cases: CAS-6, CAS-10 that are the candidate cases to be returned.

5.4. The sorting sub-step

To perform the sorting operation, the system uses all the attributes except those which have a 'null' value on the target case or on the candidate cases. The context attributes do not have the same importance than the 'description of the difficulty' attribute.

We assigned a weight equals to 0.25 to each context attribute and a weight equals to 1 to the 'description of the difficulty' attribute. Concerning the distance between the values of a context attribute, it will be 1 if the values are the same. Otherwise, it will be 0. For the 'description of the difficulty' attribute, we use the similarity index already calculated as distance. The following table (table 4) shows the calculating of sorting indices of the cases: CASE-6 and CASE-10.

Table 4. The calculating of sorting indices of the cases: CASE-6 and CASE-10

Attribute	Value of the attribute		Distance	Weight	Distance * weight
	Target case	CASE-6			
Description	-	-	0.204	1	0.204
Domaine	Informatique	Electronique	0	0.25	0
Index of sorting:					0.204

Attribute	Value of the attribute		Distance	Weight	Distance * weight
	Target case	CASE-10			
Description	-	-	0.34	1	0.34
Domaine	Informatique	Informatique	1	0.25	0.25
Index of sorting:					0.59

6.CONCLUSION

In this paper, we presented the context of our research, the kernel and the RD layer of the CCE-MAETIC, and we made a focus on the retrieve phase of the CBR cycle. Finally we presented the experimental validation of the RD layer.

The RD layer of the environment has been designed according to the CBR reasoning process in order to facilitate the transfer of know-how related to the implementation of MAETIC. Indeed, this layer provides a set of cases considered similar to the target case that would be reused to assist the teacher community.



However, the cases returned by this layer, may not meet the needs of the teacher who seeks to resolve a difficulty. In this case, he is brought to solicit the COP MAETIC in order to help him solve this difficulty. This process should result in a set of data exchanges (interactions) between the teacher and other community members. Our next task consist to model these interactions and to adapt them to the context of community members. This point will be subject of future communication.

ACKNOWLEDGMENTS

To our god, who guided us through the way.

To our friends and families, they were a great source of support and encouragement; we thank them all and wish them all the best in their lives.

To every person gave us something to light our pathway, we thank them for believing in us.

REFERENCES

- [1] Aamodt A. and Plaza E., 1994. Case Based Reasoning: Foundational Issues, Methodological Variations and System Approach. *AI Communications*, vol. 7, No. 1, pp. 39-59.
- [2] Belmeskine R. and al, 2012. Community Computer Environment supports a COP, CBR approach to solve difficulties. *IEEE International Colloquium on Information Science and Technology (CIST'2012)*. Fez, Morocco.
- [3] Belmeskine R. and al, 2012. Environnement Informatique Communautaire support d'une CoP pour l'aide à la résolution de difficultés. *8ème Colloque Technologies de l'Information et de la Communication pour l'Enseignement (TICE 2012)*. Lyon, France, pp. 31-43.
- [4] Charlier B. and al, 2007. Designing services for CoPs : first results of the PALETTE project. TEL-CoPs'07: 2nd International Workshop on Building Technology Enhanced Learning Solutions for Communities of Practice. Crete, Greece.
- [5] Chikh, A. and Berkani, L., 2010. Communities of practice of e-learning, an innovative learning space for e-learning actors. *Procedia-Social and Behavioral Sciences*. Vol. 2, No. 2, pp. 5022-5027.
- [6] Dillenbourg P. and al, 2003. *Communautés virtuelles d'apprentissage : e-jargon ou nouveau paradigme ?*. Pédagogies.Net, l'essor des communautés virtuelles d'apprentissage, Presse de l'Université du Québec.
- [7] Drupal, 2013. *Official homepage of DRUPAL*, retrieved 15 April 2013, <http://drupal.org/>.
- [8] Garrot E., 2008. Plate-forme support à l'Interconnexion de Communautés de Pratique (ICP) - Application au tutorat avec TE-Cap. PhD thesis. Institut National des Sciences Appliquées de Lyon, Lyon, France.
- [9] Helms, R.W. and al, 2006. Application of a Community of Practice to improve knowledge sharing in offshoring relations. Proceedings of the of the 7th International Workshop on Theory and Applications of Knowledge Management (TAKMA): IEEE Computer Society. Los Alamitos, USA, pp. 161-165.
- [10] Lamontagne L. and Lapalme G., 2002. Raisonnement à base de cas textuel – état de l'art et perspectives futures. *Revue d'intelligence artificielle*, Vol. 16, No. 3, pp. 339-366.
- [11] Lave, J. and Wenger, E., 1991. *Situated learning: Legitimate peripheral participation*. Cambridge University Press, Cambridge, UK.
- [12] Lewis D. D., 1992. Representation and Learning in Information Retrieval. *Technical Report UMCS-1991-093*, Department of Computer Science, University of Massachusetts.
- [13] Quenu-joiron C. and Lecllet D., 2010. How to Instrument a Community of Practice Dedicated to Project Based Pedagogy tutors : a Solution Based on Case Based Reasoning . *10th IEEE International Conference on Advanced Learning Technologies (ICALT 2010)*. Sousse, Tunisia, pp. 344-348.
- [14] Salton G. and al, 1975. A Vector Space Model for Automatic Indexing. *Communications of the ACM*, Vol.18, No.11, pp.613-620.
- [15] Salton G. and Buckley C., 1988. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, Vol. 24, pp.513-523
- [16] Salton G., 1971. *The SMART Retrieval System - Experiments in Automatic Document Processing*. Prentice Hall, Upper Saddle River, New Jersey, USA.
- [17] Shannon C. E., 1948. A mathematical theory of communication. *Bell system technical journal*, Vol. 27, pp. 379-423.
- [18] Talon B. and Lecllet D., 2008. Dispositif pédagogique pour un apprentissage de savoir-faire : Dispositif pédagogique e-mallette MAETIC. *Revue internationale des technologies en pédagogie universitaire*, Vol. 5, No. 2, pp 58-74.
- [19] Vaast E., 2002. Les communautés de pratique sont-elles pertinentes ?. *XIème Conférence annuelle de l'AIMS*. Paris, France.
- [20] Wenger, E., 1998. *Communities of practice : Learning, meaning, and identity*. Cambridge University Press, Cambridge, UK.
- [21] Wenger, E., 2001. *Supporting Communities of Practice : A Survey of Community Oriented-Technologies*. Available at: <http://www.ewenger.com/tech/index.htm> (Accessed: 1 May 2013).



[22] Wenger, E., 2006. *Communities of Practice: A brief introduction*. Available at: <http://www.ewenger.com/theory/index.htm> (Accessed: 1 May 2013).

[23] Wenger E. and al, 2002. *Cultivating Communities of Practice*. Harvard Business School Press, Boston, USA.

