ITS evaluation in classroom: the case of AMBRE-AWP

Sandra Nogry, Stéphanie Jean-Daubias and Nathalie Duclosson

LIRIS
Université Claude Bernard Lyon 1 - CNRS
Nautibus, 8 bd Niels Bohr, Campus de la Doua
69622 Villeurbanne Cedex
FRANCE
{Sandra.Nogry, Stephanie.Jean-Daubias,
Nathalie.Guin-Duclosson}@liris.cnrs.fr

Abstract

This paper describes the evaluation of an Intelligent Tutoring System (ITS) designed within the framework of the multidisciplinary AMBRE project. The aim of this ITS is to teach abstract knowledge based on problem classes thanks to the Case-Based Reasoning paradigm. We present here AMBRE-AWP, an ITS we designed following this principle for additive word problems domain and we describe how we evaluated it. We conducted first a pre-experiment with five users. Then we conducted an experiment in classroom with 76 eight-year-old pupils using comparative methods. We present the quantitative results and we discuss them using results of qualitative analysis.

Keywords: Intelligent Tutoring System evaluation, learning evaluation, additive word problems, teaching methods, Case-Based Reasoning

Introduction

This paper describes studies conducted in the framework of the AMBRE project[1]. The purpose of this project is to design Intelligent Tutoring Systems (ITS) to teach methods. Derived from didactic studies, these methods are based on a classification of problems and solving tools. The AMBRE project proposes to help the learner to acquire a method following the steps of the Case-Based Reasoning (CBR) paradigm.
We applied this principle to the additive word problems domain. We implemented the AMBRE-AWP system and evaluated this system with eight-year-old pupils in different manners.
In this paper, we first present the AMBRE principle. Then, we describe its application to additive word problems and two experiments carried out with eight-year-old pupils in laboratory and in classroom to evaluate the AMBRE-AWP ITS.

[1] **Acknowlegments:** This research has been supported by the interdisciplinary program STIC-SHS « Société de l'Information » of CNRS

The AMBRE project

The purpose of the AMBRE project is to design an ITS to help learners to acquire methods using Case-Based Reasoning [4].
The methods we want to teach in the AMBRE project were suggested by mathematic didactic studies [12] [15]. In a small domain, a method is based on a classification of problems and of solving tools. The acquisition of this classification enables the learner to choose the solving technique that is best suited to a given problem to solve.
However, in some domains, it is not possible to explicitly teach problem classes and solving techniques associated with those classes. So, the AMBRE project proposes to enable the learner to build his own method using the case-based reasoning paradigm.
Case-Based Reasoning [7] can be described as a set of sequential steps (elaborate a target case, retrieve a source case, adapt the source to find the target case solution, revise the solution, store the case). The CBR paradigm is a technique that has already been used in various parts of ITS (e.g. learner model, diagnosis). The closest application to our approach is Case-Based Teaching [1] [9] [13]. Systems based on this learning strategy present a close case to the learner when (s)he encounters difficulties in solving a problem, or when (s)he faces a problem (s)he never came across before (in a new domain or a new type).
In the AMBRE project, CBR is not used by the system, but proposed to the learner as a learning strategy. Thus, in order to help the learner to acquire a method, we propose to present him a few typical worked-out examples (serving as case base initialization). Then, the learner is assisted in solving new problems. The environment guides the learner's solving of the problem by following each step of the CBR cycle (Fig. 1): the learner reformulates the problem in order to identify problem structure features (elaboration of the CBR cycle). Then, (s)he chooses a typical problem (retrieval). Next, (s)he adapts the typical problem solution to the problem to solve (adaptation). Finally, (s)he classifies the new problem (storing). The steps are guided by the system, but done by the learner. In the AMBRE ITS, revision is included as a diagnosis of learner responses in each step of the cycle.

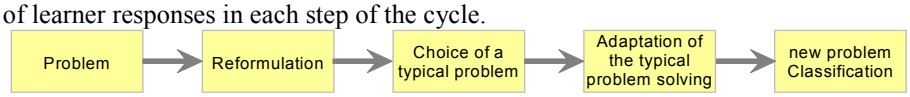

Fig. 1: The CBR cycle adapted to the AMBRE project.

The design process adopted in the AMBRE project is iterative, it is based on the implementation of prototypes that are tested and then modified This design satisfied the preoccupation with validating multidisciplinary design choices and detecting problems of use as early as possible.
Before the AMBRE design, the SYRCLAD solver [5] was designed to be used in ITS. SYRCLAD solves problems according to the methods we want to teach.
To begin the AMBRE design, we specified the objective of the project (to learn methods) and the approach to be used (CBR approach). Then we developed a first simple prototype (AMBRE-counting) for the numbering problems domain (final scientific year level, 18 year-old students). This prototype implemented the AMBRE principle with a limited number of problems, and limited functionalities (the Artificial Intelligence modules were not integrated). This prototype was evaluated in classroom

using experimental method of cognitive psychology to assess the impact of the CBR paradigm on method learning. The results did not show significant learning improvement using the AMBRE ITS. Nevertheless, we identified difficulties experienced by learners during the system use [4]. These results and complementary studies of cognitive psychology moved us to propose recommendations and new specifications.
After that, we implemented a system for additive word problem solving (AMBRE-AWP) taking into account the previous recommendations and specifications. This system includes a new interface, the SYRCLAD solver, and help and diagnosis functionalities.
This system was evaluated by developers and teachers, and used by children in laboratory. Then it was used by pupils in classroom.
In next sections, we present in more details AMBRE-AWP and we describe the evaluation of the system.

AMBRE-AWP : an ITS to solve additive word problems

AMBRE-AWP is an ITS for additive word problem solving based on the AMBRE principle. We chose additive word problems domain because this difficult domain for children is suitable to AMBRE principle. Learners have difficulties to visualize the problem situation [3]. Didactic studies proposed additive word problems classes [17] identifying problem type (add, change, compare) and the place of the unknown that can help learners to visualize the situation. Nonetheless, it is not possible to teach these classes explicitly. AMBRE principle might help the learner to identify the problem's relevant features (the problem class).
These problems are studied in primary school. Thus we adapted the system to be used individually in classroom in primary school by eight-year-old pupils.
According to the AMBRE principle, AMBRE-AWP presents examples to learner and then guides him following the steps described below.
Reformulation of the problem: once the learner has read the problem to solve (e.g. "Julia had 17 cookies in her bag. She ate some of them during the break. Now, she has 9 left. How many cookies did Julia eat during the break?"), the first step consists in reformulating the problem. The learner is asked to build a new formulation of the submitted problem identifying its relevant features (i.e. problem type and unknown place). We chose to represent problem classes by diagrams that we adapted from didactic studies [17] [18]. The reformulation no longer has most of the initial problem's surface features, and becomes a reference for the remainder of the solving.
Choice of a typical problem: the second step of the solving consists for the learner in comparing the problem to be solved with the typical problems by identifying differences and similarities in each case. Typical problems are represented by their wording and their reformulation. The learner should choose the problem that seems the nearest to the problem to be solved, such nearness being based on reformulations. By choosing a typical problem, the learner implicitly identifies the class of the problem to be solved.

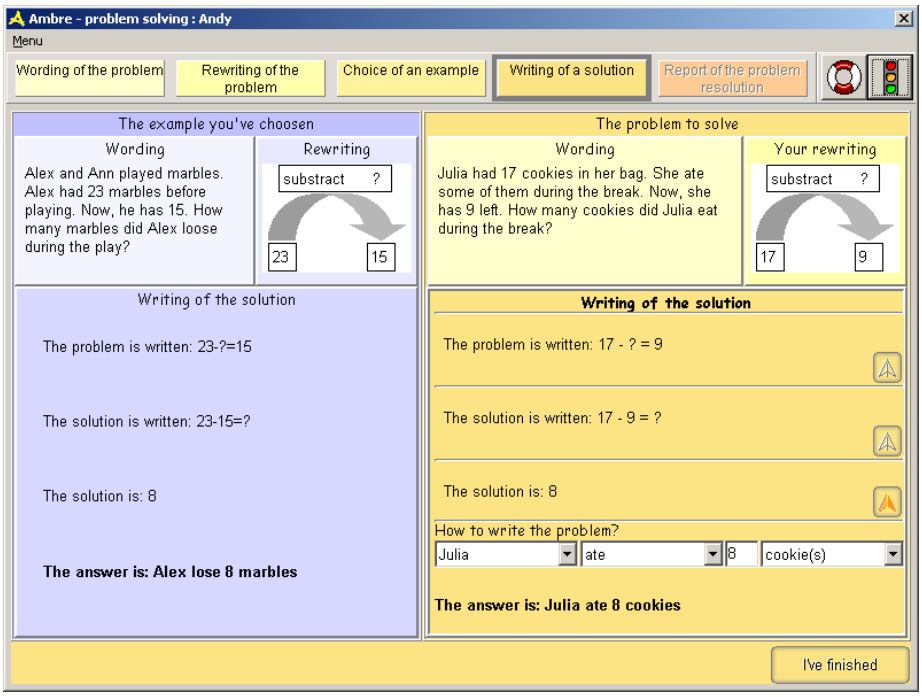

Fig. 2: Adaptation step in AMBRE-AWP (English translation of the French interface).

Adaptation of the typical problem solution to the problem to be solved: in order to write the solution, the learner should adapt the solution of the typical problem he chose in the previous step to the problem to be solved (Fig. 2). The solution writing consists first in establishing the equation corresponding to the problem. Then, the learner writes how to calculate the solution and then calculates it. Finally, (s)he constructs a sentence to answer the question. If the learner uses the help functionality, the system can assist the adaptation by outlining with colors similarities between the typical problem (Fig. 2: left side) and the problem to solve (Fig. 2: right side).

Classification of the problem: first, the learner can read the report of the problem solving. Then, he has to classify the new problem by associating it with a typical problem that represents a group of existing problems of the same class. During that step, the learner should identify the group of problems associated with the solved problem.

AMBRE-AWP evaluation with eight-year-old pupils

After the implementation, AMBRE-AWP was evaluated with pupils. To evaluate a system, Senach [16] distinguishes two aspects: the usability and the utility of the system. Usability concerns the capacity of the software to allow the user to reach his objectives easily. Utility deals with the adequacy of the software to the high level objectives of the customer. In the case of ITS, the user is the learner and the customer

is the teacher or the "educational system". So, we must take into account learner specificity in the usability evaluation. The high level objective of ITS is learning. So, the evaluation of the system utility concerns the evaluation of the learning. In our case, we have to evaluate the method learning. If usability can be evaluated with classical methods developed in Human Computer Interaction (HCI) domain, learning evaluation requires specific methods.
In this section, we present the AMBRE-AWP evaluation with eight-year-old children. We first describe a pre-experiment in laboratory, which enabled us to evaluate usability. This pre-experiment moved us to modify of the system. Then, we present evaluation of AMBRE-AWP utility in classroom.

Pre-experiment in laboratory

We evaluated AMBRE-AWP in a pre-experiment in order to observe the appropriateness of the system to the learners and to identify usability problems.
Due to the specificity of the learners (young children, beginner readers, not very familiar with computer), we chose to use one to one testing [8]: we observed individually eight-year-old learners using AMBRE-AWP in order to detect the main usability problems. They had to solve two additive word problems with the system during 45 minutes. During the use of the system, we observed interactions between the children and the system and we recorded what users said. Then, learners filled up a short questionnaire that let us to know if they liked mathematics, if they are familiar with computer use and their satisfaction.
In order to evaluate AMBRE-AWP usability, we referred to existent ergonomic criteria. Among these criteria we chose seven criteria proposed by Bastien & Scapin [2], Nielsen [11] and Schneiderman [14] that are adapted [6] to observe ITS usability: Learnability (how do users understand the system use?), general understanding (do users understand the software principle?), effectiveness (are there some interface elements that lead to systematic errors?), error management (are there ergonomic problems which lead to errors?), help management (do users use the help functionality?), cognitive load and satisfaction
We observed five users; all were familiar with computer use (regular use at school or at home) and liked mathematics. Some of them were poor readers.
First, as we expected, observations showed that users passed a lot of time to discover interface elements (e.g.: list-box). Although users encountered difficulties to use the system during the first problem resolution, these difficulties disappeared during the second problem resolution. So, the interface use seemed to be time consuming but well understood. The general understanding of the system seemed to be difficult: users did not understand well the AMBRE principle and the link between solving steps. Moreover, we observed cognitive overload during the worked-out examples presentation and the adaptation step. Furthermore, in the adaptation step (Fig. 2), learners had difficulties to write how to calculate the solution. Teachers confirmed that this sub-step was not adapted to the arithmetical knowledge of the target users.
The observation of the help functionality use showed that help was often used. Nevertheless, children did not well understand help and error messages.

Finally, the questionnaire analysis showed that four users among five were satisfied and consider AMBRE-AWP pleasant to use.
We take into account these results to adapt AMBRE-AWP to eight-year-old users capabilities, modifying the system. For example, in order to facilitate the system learnability, we chose to replace the tutorial with a demonstration explaining the AMBRE principle and showing how to use the interface during the first session; to reduce cognitive load, we modified the examples presentation. Moreover, we deleted the adaptation sub-step, which was not adapted to learners of this age.

Learning evaluation

After the pre-experiment, we evaluated the utility of the modified system measuring the impact of AMBRE-AWP on method learning for additive word problems.
More precisely, we were interested in knowing if AMBRE-AWP has an impact on the learner ability to identify the class of a problem and if the expected impact of AMBRE-AWP is due to CBR approach or if it is only due to problem reformulation with diagram.
For that, we used experimental method [8]. We compared AMBRE-AWP use with the use of two control prototypes. The experiment was conducted in classroom with 76 eight-year-old pupils divided in six groups in order to reproduce actual use conditions. During six weeks, each group worked in computer classroom and used the software during half an hour per week. Each child used individually the software. We measured the learning outcomes with different tasks and we completed these data with a qualitative approach.

Evaluation paradigm
We compared three systems: the AMBRE-AWP ITS and two control systems. The whole system, AMBRE-AWP, guides the solving toward the CBR cycle according to the AMBRE principle.
The first control system, the "reformulation and solving system" presents worked-out examples and guides the learner to solve the problem. The learner reformulates the problem and then writes the solution. Finally, he can read the problem report. In contrast with AMBRE-AWP, this system does not propose to choose and to use a prototypical example. The aim of this control system is to verify the impact of reformulation with diagrams on learning.
The second control system, the "simple solving system", proposes to find the problem solution directly. Once the learner has read the worked-out examples and the problem to be solved, he writes the solution. Finally, he can read the problem report. Contrary to the AMBRE-AWP ITS, there is no reformulation and the step of the choice of typical problem. As this system has fewer steps than the others, learners have to make another task after problem solving so that all groups solve an equivalent number of problems. This task consists in reading a problem wording and finding pertinent information in this text (a number) to answer a question.
In each of the three pupils classes, one group uses AMBRE-AWP and the other group uses of the other control systems. Learners are assigned to groups according to their mathematical level so that groups are equivalent. In order to measure the learning

outcomes, we use a "structure features detection task", a problem solving task and an "equation writing task".

"Structure features detection task" consists in reading a first problem, and then choosing between two problems the one that is solved like the first problem. In this task, we manipulate unknown place, problem type and surface features. This task enables to evaluate the learner ability to identify two problems that have the same structure features whatever the surface features and the difficulty of the problem are.

Problem solving task is a paper and pencil task. It consists in solving six problems: two problems close to problems presented by the system ("easy problems") and four problems that content non pertinent data for the resolution ("difficult problems"). This task enables to evaluate the impact of the system on paper and pencil task with simple and difficult problems.

In the "Equation writing task" we presented a diagram representing a problem class. The learner task consisted in typing the equation corresponding to the diagram (filling up boxes with numbers and operation). This task allows us to test the learner ability to associate the corresponding equation with the problem class (represented by diagrams). This task is realized only by groups that made the reformulation step (the AMBRE-AWP group and the "Reformulation and solving system" group).

The experimental design we adopt is an *interrupted-time series design*: we present the problem solving task as pre-test, after the fourth system use, as post-test and as delayed post-test one month after the last system use. The "structure features detection task" is presented after each system use; the "equation writing task" is presented after the fifth system use and as post-test.

In order to complete these data, we adopt a qualitative approach [8]. Before the experiment, we made an "a priori" analysis in order to highlight the various strategies usable by learners who solve problems with AMBRE-AWP. During the system use, we noticed all questions asked. Moreover, we observed the difficulties encountered by learners, the interactions between the learners and the interactions between the learners and the persons that supervise the sessions. In post-test, the learners filled up a questionnaire in order to take into account their satisfaction and remarks. Finally, we analysed the use traces in order to identify the strategies used by learners, to highlight the most frequent errors and to identify the steps that cause difficulties to learners. With these methods, we would like to identify difficulties encountered by learners and want to take into account the complexity of the situation.

Results

In this section, we present the quantitative results and we discuss these results using qualitative results.

With the problem solving task, we performed an analysis of variance on performances with groups (AMBRE-AWP, simple solver system, Reformulation and solving system) and tests (4 tests) as variables. Performances in pre-test are significantly lower than performance of the other tests ($F(3,192)=18.1$; $p<0.001$). There is no significant difference between tests performed after the fourth system use, as post-test and as delayed post-test one month after the last system use. There is no significant differences between groups ($F(2,64)=0.12$; $p=0.89$) and no interaction between group and sessions ($F(6,192)=1.15$; $p=0.33$). With the "structure features detection task", there is no significant difference between the AMBRE-AWP group and the other

groups (Chi^2 (dl=1)=0.21; p= 0.64). Even at the end of the experiment, surface features interfere with structure feature in problem choice. The "equation writing task" shows that learners that use AMBRE-AWP and "Reformulation and solving system" are both able to write the right equation corresponding to a problem class represented by a diagram in fifty percent of the cases. Thus there is no difference between the results of the AMBRE-AWP group and the control groups for each task. The three systems equally improve learning outcomes. Results of "structure feature detection task" and "equation writing task" do not show method learning. So, these results do not validate the AMBRE principle.

The qualitative analysis allows explaining these results. First, pupils did not use AMBRE-AWP as we expected. The observation shows that when they wrote the solution, they did not adapt the typical problem to solve the problem. Secondly, learners solved each problem very slowly (means 15 minutes). As they are beginner readers, they had difficulties to read instructions and messages, and were discouraged sometimes to read them. Besides, they met difficulties during reformulation and adaptation steps because they did not identify well their mistakes and they did not master arithmetic techniques. Thirdly, the comparison between "simple solving system" and AMBRE-AWP is questionable. Indeed, despite the additional task, the "simple solving system" group resolved significantly more problems than the AMBRE-AWP group (means 9 problems vs. 14 problems during the 6 sessions, $F(1, 45) = 9.7$; $p<0.01$). Moreover assistance required by pupils and given by persons that supervised sessions varied with groups. With AMBRE-AWP, questions and assistance often consisted in reformulating help and diagnosis messages. Whereas, in the simple solving system it consisted in giving mathematic helps sometimes comparable to AMBRE-AWP reformulation. So, even if AMBRE principle has an impact on learning, the difference between number of problems solved by AMBRE-AWP group and "simple solving system" group and the difference of assistance could partly explain that these two groups have similar results.

Thus, the quantitative results (no difference between groups) can be explained by three reasons. First, pupils did not use prototypical problems to solve their problem. As we expected that the choice and adaptation of a typical problem could facilitate analogy between problems and favour method learning, it is not surprising that we do not observe method learning. Secondly, learners solved each problem slowly and they were confronted with a lot of difficulties (reading, reformulation, solution calculating) all over the AMBRE cycle. These difficulties probably disrupt their understanding of the AMBRE principle. Third, there are methodological issues due to the difficulty to use comparison method in real word experiments because it is not possible to control all factors. A pre-test of the control system should decrease these difficulties but not suppress them. These methodological issues confirm our impression that it is necessary to complete experimental method with qualitative approach to evaluate an ITS in real word [10].

These qualitative results show that AMBRE-AWP is not well adapted for eight-year-old pupils. However, questionnaire and interviews showed that a lot of pupils were enthusiastic to use AMBRE-AWP (more than the "simple solver system"); they appreciated to reformulate the problem with diagrams.

Conclusions and Prospects

The framework of the study described in this paper is the AMBRE project. This project relies on the CBR solving cycle to have the learner acquire a problem solving method based on a classification of problems. We implemented a system based on the AMBRE principle for additive word problems solving (AMBRE-AWP). We evaluated it with eight-year-old pupils. In the first experiment, we observed five children in laboratory, in order to identify some usability problems and to verify the adequacy of the system with this type of users. Then, we realized an experiment in classroom during six week with 76 pupils. We compared the system with two control systems to assess the impact of the AMBRE principle on method learning. Results show performances improvement between pre-test and post-test but no difference between the AMBRE-AWP group and the other groups. Thus the AMBRE-AWP system improves learning outcomes but not more than other systems. These results cannot allow us to validate the AMBRE principle. The qualitative results show that learners did not use the system like we expected it. They construct the solution without adapting the typical problem solution. Moreover, they had difficulties like reading, and calculating that slowed down the problem solving.

This experiment leads us to modify some aspects of the system. We modified the diagnosis messages so that they are more understandable for primary school pupils. Moreover, in order to reduce the difficulties due to reading, we consider integrating to AMBRE-AWP a text-to-speech synthesis system in order to present the diagnosis messages and instructions.

Furthermore, as that AMBRE-AWP is too complex for eight-year-old pupils, we are trying to identify learners for whom AMBRE-AWP is more appropriate. At present, we are testing the system with twenty nine-year-old pupils in order to evaluate if they have less difficulties than eight-year-old pupils and if problems are adapted to them. If this pre-test is positive, we will evaluate the AMBRE principle with them.

Besides, in collaboration with teachers, we design simpler activities preparatory to AMBRE-AWP within the reach of young pupils in order to acquire capabilities used in AMBRE-AWP. For example, we propose activities that develop the capability to identify relevant features in the problem wording. We also develop activities that highlight the links between the wording of the problem, its reformulation and its solving showing how a modification on the wording acts on its reformulation, how a modification on the reformulation acts on its wording, and what are the consequences of these modifications on the solving.

Finally, we propose two long-term prospects. We study the possibility to propose AMBRE-AWP to adults within a literacy context, using new story types in the wordings problems. We are also designing an environment for teachers enabling them to customize the AMBRE-AWP environment and to generate the problems they wish their pupils to work on with the system.

References

1. Aleven, V. & Ashley, K.D.: Teaching Case-Based Argumentation through a Model and Examples - Empirical Evaluation of an Intelligent Learning Environment. Artificial Intelligence in Education, IOS Press (1997), 87-94.
2. Bastien, C. & Scapin, D.: Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. In RT n°156, INRIA, (1993).
3. Greeno, J.G. & Riley, M.S.: Processes and development of understanding. In metacognition, motivation and understanding, F.E. Weinert, R.H. Kluwe Eds (1987), Chap 10, 289-313.
4. Guin-Duclosson, N., Jean-Daubias, S. & Nogry, S.: The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In proceedings of ITS, Biarritz, France: Springer (2002), 782-791.
5. Guin-Duclosson, N.: SYRCLAD: une architecture de résolveurs de problèmes permettant d'expliciter des connaissances de classification, reformulation et résolution. Revue d'Intelligence Artificielle, vol 13-2, Paris : Hermès (1999), 225-282
6. Jean, S.: Application de recommandations ergonomiques : spécificités des EIAO dédiés à l'évaluation. In proceedings of RJC IHM 2000 (2000), 39-42
7. Kolodner, J.: Case Based Reasoning. San Mateo, CA: Morgan Kaufmann Publishers (1993).
8. Mark, M. A., & Greer, J. E.: Evaluation methodologies for intelligent tutoring systems. Journal of Artificial Intelligence in Education, vol 4.2/3 (1993), 129-153.
9. Masterton, S.: The Virtual Participant: Lessons to be Learned from a Case-Based Tutor's Assistant. Computer Support for Collaborative Learning, Toronto (1997), 179-186.
10. Murray, T.:. Formative Qualitative Evaluation for "Exploratory" ITS research. Journal of Artificial Intelligence in Education, vol 4(2/3, (1993), 179-207.
11. Nielsen, J.: Usability Engineering, Academic Press (1993).
12. Rogalski, M.: Les concepts de l'EIAO sont-ils indépendants du domaine? L'exemple d'enseignement de méthodes en analyse. Recherches en Didactiques des Mathématiques, vol 14 n° 1.2 (1994), 43-66.
13. Schank, R. & Edelson, D.: A Role for AI in Education: Using Technology to Reshape Education. Journal of Artificial Intelligence in Education, vol 1.2 (1990), 3-20.
14. Schneiderman, B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA : Addison-Wesley (1992).
15. Schoenfeld, A.: Mathematical Problem Solving. New York: Academic Press (1985).
16. Senach, B.: L'évaluation ergonomique des interfaces homme-machine. L'ergonomie dans la conception des projets informatiques, Octares editions (1993), 69-122.
17. Vergnaud, G.: A classification of cognitive tasks and operations of the thought involved in addition and substraction problems. Addition and substraction: A cognitive perspective, Hillsdale: Erlbaum (1982), 39-58.
18. Willis, G. B. & Fuson, K.C.: Teaching children to use schematic drawings to solve addition and subtraction word problems. Journal of Educational Psychology, vol 80 (1988), 190-201.