ITS evaluation in classroom: the case of the AMBRE ITS

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

Article soumis le 15 avril 2004 à la conférence ITS 2004

1 Introduction

This paper describes studies conducted in the framework of the Ambre project1. 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.

2 The AMBRE project

The purpose of the AMBRE project is to design an ITS to help learners to learn methods [14]. In a domain, method is based on didactic studies. The AMBRE principle is to use the Case-Based Reasoning paradigm to use problems already met during learning, in order to facilitate the method learning.

In this section, we first present what is a method. Then, we describe how we use the CBR paradigm in the AMBRE project. Last, we present the design cycle of the system.

2.1 Method learning

The methods we want to teach in the AMBRE project were suggested by mathematic didactic studies [26] [29]. In a small domain, methods are based on a classification of problems and of solving tools. The acquisition of these classifications enables the learner to choose the solving technique that is best suited to a given problem to solve.

The AMBRE project aims at designing an ITS the purpose of which is to have the learner acquire a method.

However, in some domains, it is not possible to teach explicitly 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.

2.2 Guiding learning using the Case–Based Reasoning paradigm approach

Case-Based Reasoning [20] can be described as a set of sequential steps (elaborate, retrieve, adapt, revise, store) that is often represented by a cycle [1] [23] (Fig. 1).

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

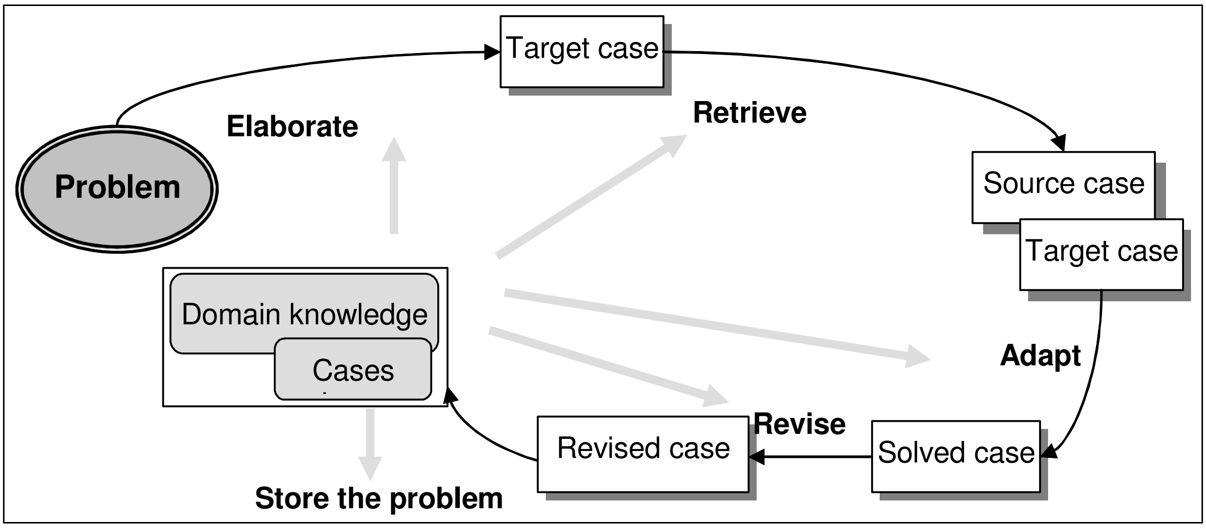

Fig. 1. The CBR cycle

The CBR paradigm is a technique that has already been used in various parts of ITS. Modeling the learner's knowledge can be done using CBR [31], like the assessment of the learner [19] and the diagnostic of his or her mistakes [3]. CBR can also help selecting a learning strategy [12], or adapting a hypermedia to the learner [16] [10] [32] by comparing the learner's model with other learners' assessments (forming a case base).

The closest application to our approach is Case-Based Teaching [27] [22] [2] [5] [7] [18]. Systems based on this learning strategy present a close case to the learner when he encounters difficulties in solving a problem, or when he faces a problem he never came across before (in a new domain or a new type). In these systems, there are several levels of interactivity between the learner and the computer environment [33]. The learner can ask the system to find a similar example and to explain how that case was solved; the system can also propose a case to the learner when it seems to be necessary. In these systems, cases represent the experience of an expert or other learners. Thus, Case-Based Teaching systems use CBR as a technique to find a case that can help the learner.

In the AMBRE project, CBR is not used by the system, but proposed to the learner as a learning strategy. The learner has to retrieve a case himself. Moreover, the complete CBR cycle is used from elaboration to storing.

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. 2): the learner reformulates the problem in order to identify problem structure features (elaboration). Then, the learner retrieves a typical problem from the case base (retrieval). Next, the learner adapts the typical problem solution to the problem to solve (adaptation). Finally, the learner stores the new problem in the case base (storing). The steps are guided by the system, but done by the learner. In the AMBRE ITS, revision is included in each step of the cycle, as it is explained in part 3.

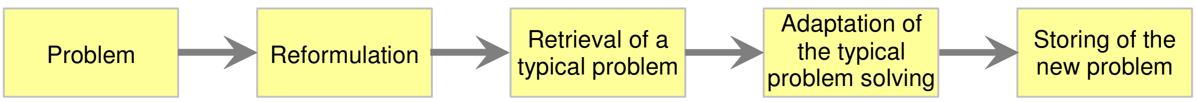

Fig. 2. The CBR cycle adapted to the AMBRE project

2.3 Design process

In this section we present the design process used in the AMBRE project. Figure 3 shows the AMBRE project development cycle adapting a diagram proposed by Bruillard & Vivet [6] that highlights users, tests and validations. The preoccupation with validating multidisciplinary design choices and detecting problems of use as early as possible leads us to the adoption of an iterative design process based on the implementation of prototypes that are tested and then modified.

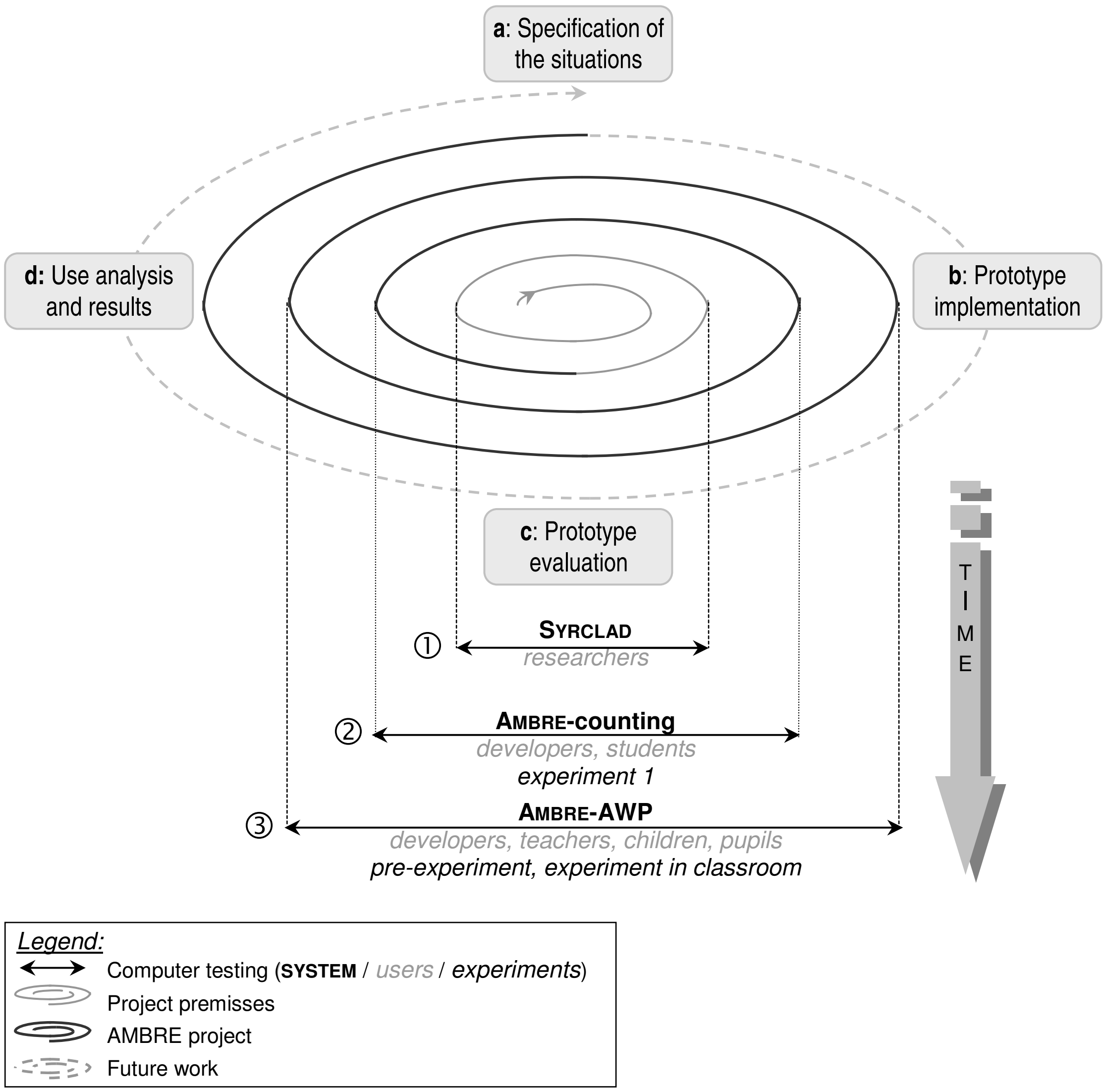

Fig. 3. The AMBRE project design cycle

Before the AMBRE design, the SYRCLAD solver [15] was designed to be used in ITS. SYRCLAD solves problems according to the methods we want to teach [8]. This solver was tested by researchers (Fig. 3, 1).

To begin the AMBRE design, we specified the objective of the project (to learn methods) and the approach to be used (CBR approach) (Fig. 3, 2a). Then we developed a first simple prototype (AMBRE-counting) for the numbering problems domain (final scientific year level, 18 year-old students) (Fig. 3, 2b). 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 (Fig. 3, 2c) 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 [14]. These results and complementary studies of cognitive psychology leaded us to recommendations and new specifications (Fig. 3, 3a).

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 diagnostic functionalities.

This system was evaluated by developers and teachers, and used by children in laboratory. Then it was used by pupils in classroom (Fig. 3, 3c).

In next sections, we present in more details AMBRE-AWP and we describe the evaluation of the system.

3 AMBRE-AWP : an ITS to solve additive word problems

AMBRE-AWP is an ITS for additive word problem solving based on the AMBRE principle. This ITS is designed to be used individually in classroom in primary school by eight-year-old pupils. We chose the problems to solve and we adapted the vocabulary to these users.

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 [13]. Didactic studies proposed additive word problems classes [34] identifying problem type (add, change, compare) and place of 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 problem's relevant features (the problem class).

In this section, we present AMBRE-AWP functioning: we describe the steps that enable the learner to solve the problem.

3.1 Reformulation of the problem

In AMBRE-AWP, once the learner has observed examples and 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 to be solved. The learner is asked to build a new formulation of the submitted problem identifying the relevant features of the problem to be solved (i.e. problem type and unknown place). We chose to represent problem classes by diagrams that we adapted from didactic studies [34] [35] (Fig. 4). The reformulation no longer has most of the initial problem's surface features, and becomes a reference for the remainder of the solving.

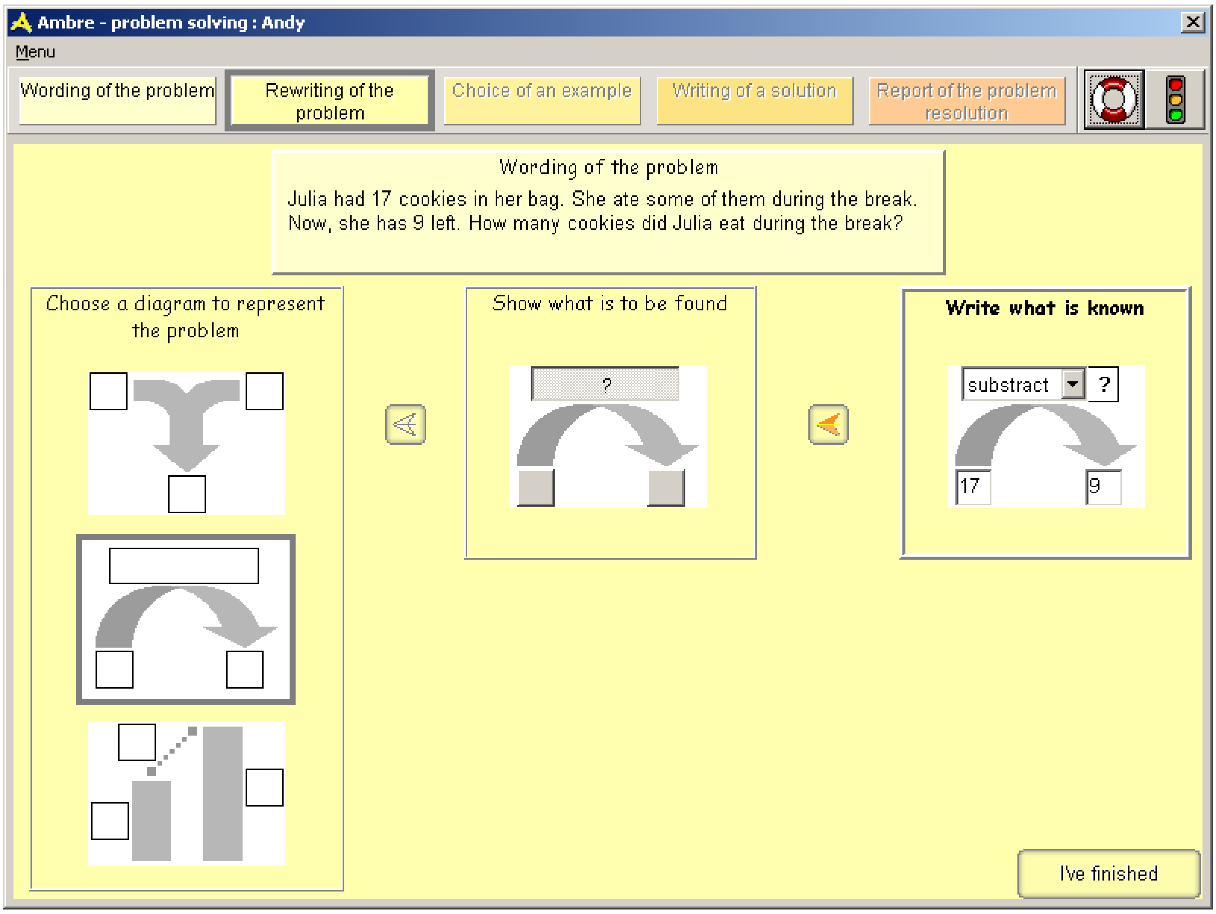

Fig. 4. Reformulation step in AMBRE-AWP

3.2 Retrieval 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. The 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 prototype problem, the learner implicitly identifies the class of the problem to be solved.

3.3 Adaptation of the typical problem solution to the problem to solve

In order to redact 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. 5). If the learner uses the help functionality, the system can assist the adaptation by outlining with colors similarities between the typical problem (Fig. 5: left side) and the problem to solve (Fig. 5: right side). 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, the learner writes a sentence to answer the question.

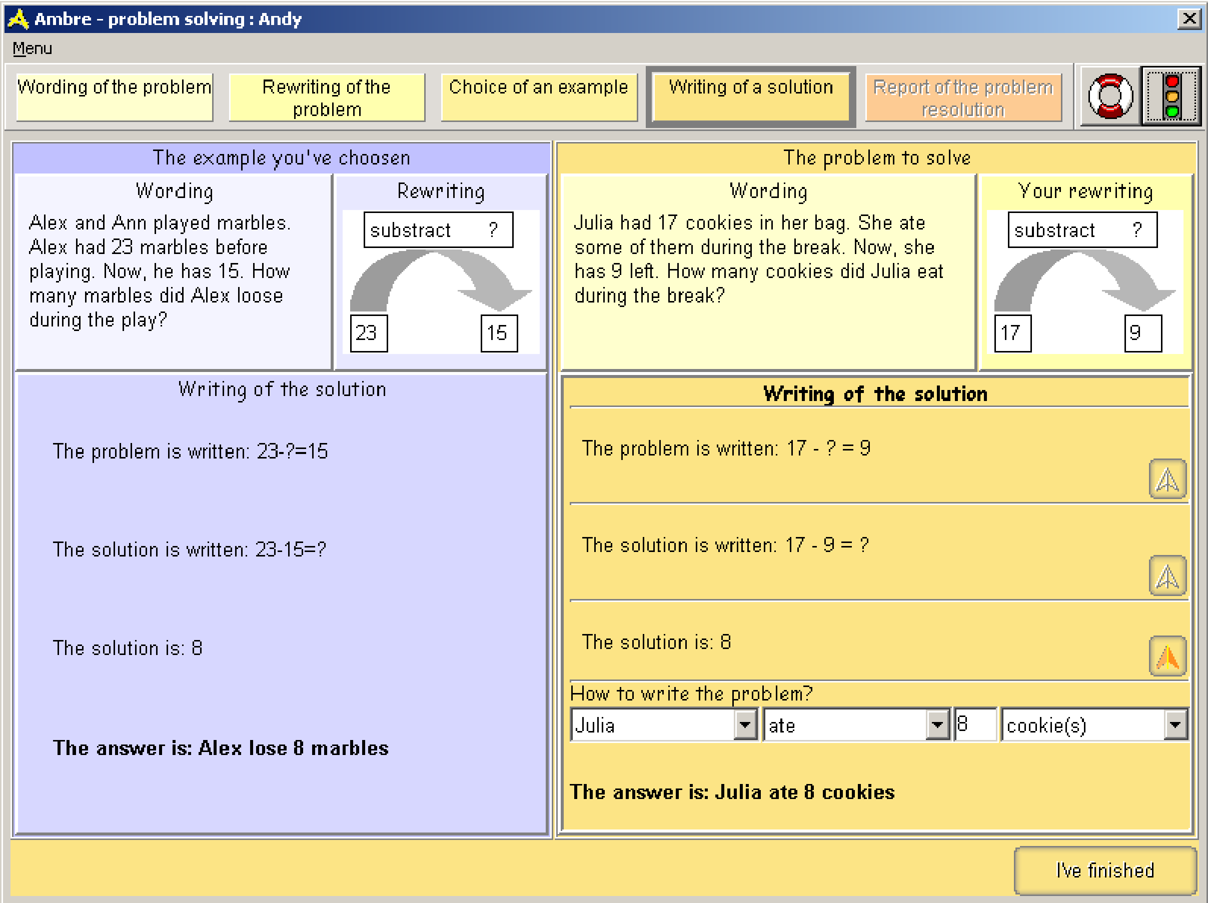

Fig. 5: Adaptation step in AMBRE-AWP

3.4 Storing of the new problem

Finally, the learner stores the new problem 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 problem to be solved.

3.5 Revision

Contrary to the classical CBR cycle, the revision is not a step in the AMBRE cycle. In our system, the revision step is replaced with a diagnostic at the end of each step. This diagnostic indicates to the learner mistakes he or she made and sometimes additional information. For instance, when an error occurs in the reformulation step, the system generates the problem wording corresponding to the reformulation made by the learner. Thus, the learner can compare the wording generated by the system with the wording of the problem to solve.

4 AMBRE-AWP evaluation with eight-year-old pupils

To evaluate a system, Senach [30] 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 using 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 leaded to modifications of the system. Then, we present evaluation of AMBRE-AWP utility in classroom, describing our objectives, the method we use and the expected results.

4.1 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 [11]: we observed in details how each learner interacted with the instructional material and then, we presented him or her a questionnaire.

4.1.1 Experimental method

According to Nielsen [24], the observation of five users allows to detect the main usability problems. So, we observed individually five eight years old learners using AMBRE-AWP. 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, gestures, non verbal behaviours, and we recorded what users said. After the use of the system, learners filled up a questionnaire (a one short in order to avoid cognitive overload). This questionnaire enabled us to know if learners liked mathematics, if they were familiar with computer use and their satisfaction.

Before the experiment, among existent usability criteria, we referred to the criteria proposed by Bastien & Scapin [4], Nielsen [25] and Schneiderman [28] and we chose criteria adapted [17] to observe AMBRE-AWP usability:

- **Learnability** (the degree to which a user interface can be learned quickly and effectively): How do users understand the system use?
- **General understanding**: do users understand the software principle?
- **Effectivness** (the degree to which an interface facilitates a user in accomplishing the task for which it was intended): is there some interface elements that lead to systematic errors?
- **Error management**: is there ergonomic problems which lead to errors? Do users understand their errors? How do they react to feedback messages?
- **Help management**: do users use the help functionality and understand the help messages?
- **Cognitive load**: is there a cognitive overload?
- **Satisfaction:** is the system pleasant to use?

4.1.2 Results

We noticed that all users we observed 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 link between solving steps and the AMBRE principle. They began to understand it only at the end of the first use.

Moreover, we observed cognitive overload during the worked-out examples presentation and the adaptation step. Furthermore, in the adaptation step (Fig. 5), learners had difficulties to write how to calculate the solution. This step was not appropriated to eight-year-old learner knowledge.

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 appropriate to learners of this age.

4.2 Learning evaluation

After the pre-experiment, we are now evaluating the utility of the modified system: we measure the AMBRE-AWP effect on method learning for additive word problems.

In this aim, we use experimental method [21]. Thus, we are comparing AMBRE-AWP use with the use of two control prototypes during six sessions in classroom. We measure outcomes learning by different ways and we complete these data with a qualitative approach.

4.2.1 Specific questions to answer

During this experiment, we want to evaluate the AMBRE-AWP impact on method learning.

More precisely, we would like to know if the AMBRE-AWP has an impact on the learner ability to identify the class of a problem (i.e. problem type and unknown place) and to establish the equation corresponding to the problem class.

We state that

- AMBRE-AWP enables to identify that two problems have the same type.
- AMBRE-AWP enables to recognize problem type and unknown place, whatever surface features and difficulty of the problem.
- AMBRE-AWP might enable the learner to find the equation corresponding to the problem class.
- The identification of problem class might help pupils to solve difficult problems.

Moreover, we want to know if the AMBRE-AWP impact is due to CBR approach or if it is only due to problem reformulation with diagram.

Furthermore, we would like to understand learning processes implied in the system use. And eventually we want to observe how learners appropriate AMBRE-AWP during a repeated use.

4.2.2 Situation of evaluation

We designed AMBRE-AWP to be used in classroom. Thus, 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 work in computer classroom and use the software during half an hour. Each child uses the software individually.

4.2.3 Evaluation paradigm

We use a comparison design between three systems: the AMBRE-AWP ITS and two control systems (Fig. 6).

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 redacts the solution. Finally, he can read the problem report. In contrast with AMBRE-AWP, this system does not propose to retrieve and 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. The system presents worked-out examples. Then, after the problem presentation, the learner redacts the solution. Finally, he can read the problem report. Contrary to the AMBRE-AWP ITS, there is no reformulation and retrieval steps. As this system has fewer steps than the others, learners have to make another task after problem solving. This task consists in finding pertinent information in a text to answer a question so that all groups solve an equivalent number of problems.

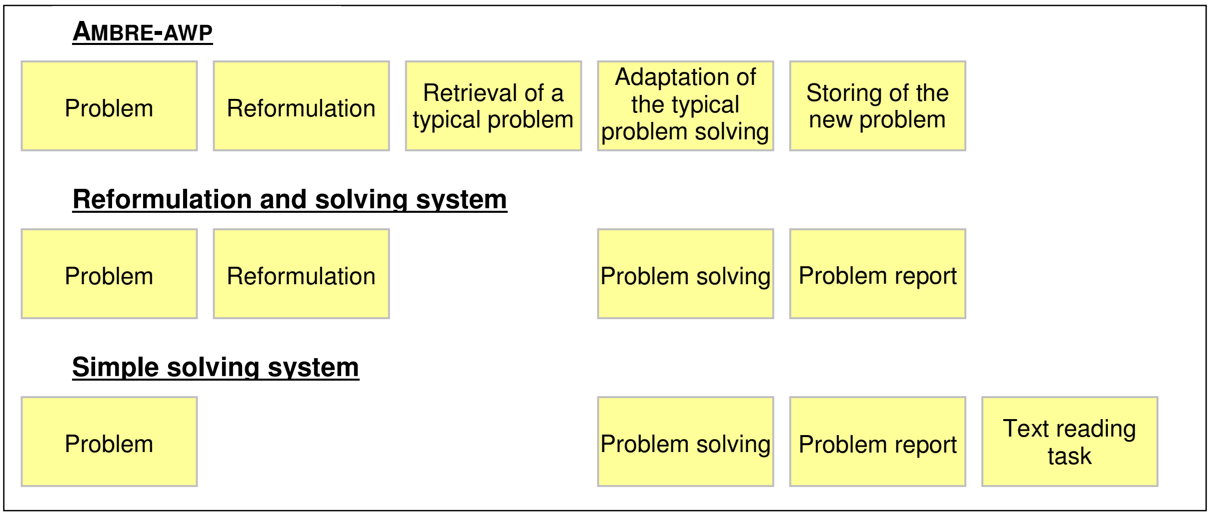

Fig. 6. The three systems used in the experiment

In each of the three pupils classes one group uses AMBRE-AWP and the other uses one of both control systems. Learners are assigned to groups according to their mathematical level so that groups are equivalents. 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 solve like the same solution than the first problem. To choose problems for this task, we manipulate unknown place, problem type and surface features. This task enables to evaluate learner ability to identify two problems that have the same structure features.

Problem solving task consists in solving six problems, two problems close to problems presented by the system (“easy problems”), two problems that content non pertinent data for the resolution (“difficult problems”) and two “transfer problems” (problems with non pertinent data and surface feature not presented by the system).

“Equation writing task” consists in writing the equation corresponding to a proposed diagram. 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 was an *interrupted-time series design* (Fig. 7): we present the problem solving task as pre-test, after the fourth system use, as post-test and differed post-test. 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.

At the end of this experiment, we expect that the use of the AMBRE-AWP system improves learning outcomes and facilitates the acquisition of problem classes:

- With the “structure features detection task”, we expect that, in post-test, the AMBRE-AWP group chooses more often the problem which has the same structure features than the group that use the “simple solving system”.
- With the problem solving task, performances in post-test may be better than in pre-test in all groups and with all problems. Moreover we might observe significant differences between the AMBRE-AWP group and the group that use the “simple solving system” for difficult problems and transfer problems solving performances.
- With the “equation writing task”, we expect that learners are able to write the right equation corresponding to a diagram.
- The comparison between the results of the “AMBRE-AWP” group and the “reformulation and solving system” group will inform us about the impact of the reformulation using diagram on learning outcomes.

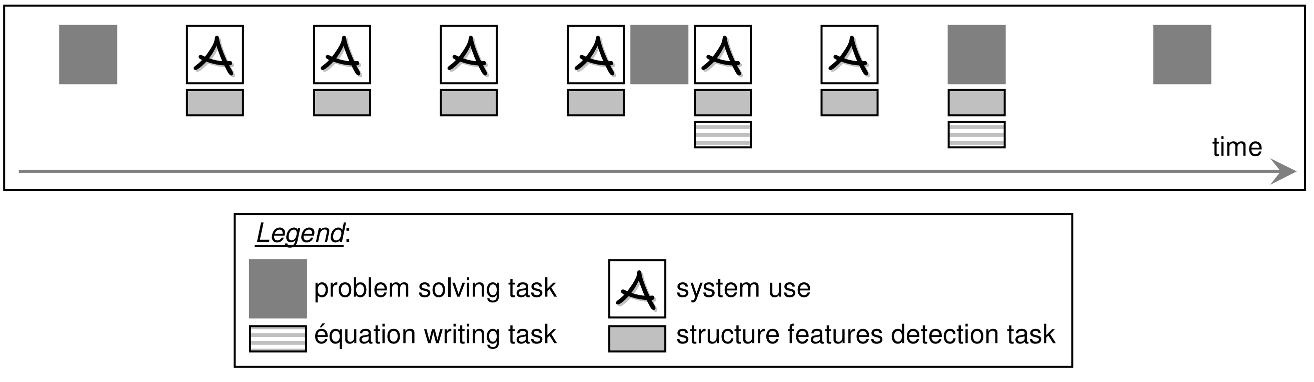

Fig. 7. Experiment design

4.2.4 Qualitative approach

The experimental method enables us to evaluate learning outcomes, but it does not allow us to take into consideration others aspects:

- We would like to understand strategies learners use to solve problems with AMBRE-AWP.
- We want to identify difficulties encounter by learners. More precisely, do these difficulties disappear during a repeated use of the system or do they persist?
- We want to take into account the complexity of the situation. Thus, we notice interactions between persons that supervise the sessions and learners and interactions between learners.

Therefore we complete our analyses using qualitative methods [21] [9].

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 notice all questions asked. Moreover, we observe the difficulties encountered by learners, the interactions between learners and the interactions between learners and persons that supervise the sessions. In post-test, the learners will fill up a questionnaire in order to take into account their satisfaction and remarks. Finally, we will analyse 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 l difficulties to earners.

4.2.5 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 [**Erreur ! Source du renvoi introuvable.**].
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. So we want to conduct another experiment with nine-year-old pupils

5 Conclusions and perspectives

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 which 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. Aamodt A. & Plaza E.: Case-based reasoning: Foundational issues, methodological variations, and system approaches. Artificial Intelligence Communications, vol 7 (1994), 39-59

2. 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 (B. du Boulay and R. Mizoguchi Eds.), IOS Press (1997), 87-94

3. Aka, M. & Frasson, C.: A New Approach on Flight Training: ASIMIL. In Proceedings of international symposium TICE 2002, Lyon, France (2002), 307–314

4. Bastien, C. & Scapin, D.: Ergonomic Criteria for the Evaluation of Human-Computer Interfaces. In RT n°156, INRIA, (1993)

5. Bélanger, S., Thibodeau, M.-A. & Aïmeur, E: Training of the Learner in Criminal Law by Case-Based Reasoning. In IEA/AIE-99:Twelve International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Cairo, Egypt (1999), 398-408

6. Bruillard, E. & Vivet, M.: Concevoir des EIAO pour des situations scolaires : approche méthodologique., Didactique et intelligence artificielle (N. Balacheff et M. Vivet Eds), La pensée sauvage éditions (1994), 273-302

7. Capus, L. & Tourigny, L.: Le raisonnement à partir de cas : une aide à la formation en analyse de sécurité routière. In Proceedings of international symposium TICE 2000, Troyes, France (2000), 227–235,

8. Delozanne, E.: Explications en EIAO : études à partir d'ÉLISE, un logiciel pour s'entraîner à une méthode de calcul de primitives. Phd, Université du Maine, Le Mans, France (1992)

9. Fasse, B.B. & Kolodner, J.L.: Evaluating Classroom Practices Using Qualitative Research Methods: Defining and Refining the Process. In Proceedings International Conference of the Learning Sciences (2000), 193-198

10. Funk P. & Conlan O.: Case-Based Reasoning to Improve Adaptability of Intelligent Tutoring Systems. Workshop on Case-Based Reasoning for Education and Training, CBRET'2002, Aberdeen, Scotland (2002), 15–23

11. Gagné, R.M., Briggs, L.J., & Wager W.W. Principles of instructional design. New York, Holt, Renhart and Winston Eds (1988)

12. Gilbert, J.E.: Case-Based Reasoning Applied to Instruction Method Selection for Intelligent Tutoring Systems. Workshop 5: Case-Based Reasoning in Intelligent Training Systems, ITS'2000, Montreal (2000), 11-15

13. 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.

14. Guin-Duclosson N., Jean-Daubias S. & Nogry S.: The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods. In proceedings of Intelligent Tutoring Systems, Biarritz, France: Springer (2002), 782-791

15. 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

16. Héraud, J.-M.: Pixed : towards the sharing and the re-use of experience to assist training. In World Conference on Educational Multimedia,Hypermedia & Telecommunications, Denver, USA (2002), 777-782

17. Jean S.: Application de recommandations ergonomiques : spécificités des EIAO dédiés à l'évaluation. In proceedings of Rencontres Jeunes Chercheurs en IHM 2000 (2000), 39-42

18. Joiron C., Leclet D.: Inciting discussions between physicans about clinical cases: The Diacom forum and its pairing module. In Proceedings of Conference on Open and Online Learning, ICOOL 2003, Ile Maurice (2003)

19. Khan, T.M.: Case-Based Evaluation for Student Modelling. Workshop 5: Case-Based Reasoning in Intelligent Training Systems, ITS'2000, Montreal (2000), 16-22

20. Kolodner, J.: Case Based Reasoning. San Mateo, CA: Morgan Kaufmann Publishers (1993)

21. 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.

22. 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

23. Fuchs B., Mille A., & Chiron. B: Operator decision aiding by adaptation of supervision strategies. In Lecture Notes in Artificial Intelligence vol 1010, First International Conference on Case-Based Reasoning, ICCBR'95, Sesimbra, Portugal, Springer-Verlag, Berlin, Germany (1995), 23-32

24. Nielsen, J.: Estimating the number of subjects needed for a thinking aloud test. International Journal of Human-Computer Studies**,** vol 41.3 (1994), 385-397

25. Nielsen J.: Usability Engineering, Academic Press (1993)

26. 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

27. 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

28. Schneiderman B.: Designing the User Interface: Strategies for Effective Human-Computer Interaction. Reading, MA : Addison-Wesley (1992)

29. Schoenfeld, A.: Mathematical Problem Solving. New York: Academic Press (1985)

30. Senach B.: L'évaluation ergonomique des interfaces homme-machine. L'ergonomie dans la conception des projets informatiques (J.-C. Sperandio Ed.), Octares editions (1993), 69-122

31. Shiri, A., Aimeur E., & Frasson, C.: SARA: A Cased-Based Student Modelling System. Fourth European Workshop on Case-Based Reasoning, Lecture Notes in Artificial Intelligence, n° 1488, Dublin, Ireland (1998), 425-436

32. Stottler R.H. & S. Ramachandran S.: A Case-Based Reasoning Approach to Internet Intelligent Tutoring Systems (ITS) and ITS Authoring. In FLAIRS Conference (1999), 181–186

33. Tourigny N., & Capus, L.: Towards Making Intelligent Training Systems Using Examples more Flexible and Reusable by Exploiting Case-Based Reasoning. Workshop 5: Case-Based Reasoning in Intelligent Training Systems, ITS'2000, Montreal (2000), 23-28

34. Vergnaud G.: A classification of cognitive tasks and operations of the thought involved in addition and substraction problems. Addition and substraction: A cognitive perspective (T.P. Carpenter, J.M. Moser & T.A. Romberg Eds), Hillsdale: Erlbaum (1982), 39-58

35. 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