AMBRE-add: An ITS to Teach Solving Arithmetic Word Problems

SANDRA NOGRY, NATHALIE GUIN* AND STÉPHANIE JEAN-DAUBIAS

LIRIS Lab., Université de Lyon
Nautibus, 43 Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France

This paper describes the design and evaluation of AMBRE-add. This Intelligent Tutoring System (ITS) was designed to teach problem classes and the solving techniques associated to those problem classes in the domain of arithmetic. AMBRE-add uses case-based reasoning as a learning strategy. It presents solved problems in each class to learners, and then guides them to use the case-based reasoning steps to solve new problems. We evaluated this system in the classroom on four classes.

Keywords: Intelligent Tutoring Systems, Case-Based Reasoning, problem solving, cognitive psychology, ITS evaluation.

INTRODUCTION

This paper describes the design and the evaluation of AMBRE-add, an Intelligent Tutoring System (ITS) that teaches problem classes in arithmetic. This ITS was designed in the framework of the AMBRE project. The aim of this project is to demonstrate that Case-Based Reasoning (CBR) can be proposed as a learning strategy by an ITS. The AMBRE project proposes to design ITSs in several domains. These ITSs will present examples to learners and then guide them in solving a new problem following the CBR steps. These systems should
facilitate the acquisition of problem classes and the solving techniques associated to these classes. We applied this principle to the design of the AMBRE-add ITS, intended to help seven- to nine-year-old children learn how to solve arithmetic word problems.

In this paper, we first describe what we want to teach with AMBRE-add. Then, we present its principle, which is to use case-based reasoning to guide the learner. Next, we describe the AMBRE-add ITS and how we evaluated it in a real-word setting.

TEACHING ARITHMETIC PROBLEM CLASSES

Research on didactics of mathematics that has studied experienced problem solvers has shown that “by the time the mathematician has finished reading the problem, the problem will have been characterized and the standard procedure for solving such problem will have been invoked. [...] The response may be a pre-packaged solution to the entire problem, if the problem is recognized as being a standard type.” (Schoenfeld 1985, p. 20). However, in some domains, the terms defining problem classes and problem-solving techniques are not used by teachers and are not known by novices. So novices have difficulty learning these problem classes.

The objective of the AMBRE project is to design computer environments using case-based reasoning to teach problem classes and solution techniques to be applied to each class. To design such an ITS, it is necessary to know the problem classes in the domain and the associated techniques. In arithmetic, such a classification already exists. Arithmetic problems describe a concrete situation such as a game of marbles: “Brad went to school with marbles. He gave thirteen of the marbles to Luke during the day. In the evening, he had fifty-six left. How many marbles did he have when he went to school?”

Riley, Greeno, and Heller (1983) suggested a classification for addition and subtraction problems based on their underlying semantic relations that draw a distinction between three main categories of problems: combine, change, and compare (Riley et al.). The classification established by Vergnaud (1982) further specifies the Riley et al. classification.

This body of research on didactics of mathematics leads to the idea that the domain of arithmetic problems is a good one for trying out an AMBRE ITS. They are problems in which modeling plays a major role, for which a classification has been established, and which are difficult for elementary school pupils. Even if we have an explicit classification of the problems and of the techniques at our
disposal, we know that it is not necessarily desirable to explicitly present the classification to the learner. In the domain of arithmetic, the terms defining the problem classes and problem-solving techniques are not used in the schools and are not known to learners. Moreover, it seems preferable for the learner to play an active part and thus to build his or her own classification.

THE AMBRE CYCLE

To assist the learner in building his or her own classification, we chose to show solved problems and next to encourage the learner to apply analogical reasoning to solve other problems. In the AMBRE project, CBR is not used by the system, but is proposed to the learner as a strategy. Thus, in order to help learners acquire problem classes and the techniques associated with those classes, they are first presented with a few typical solved problems (serving as case-base initialization). Then they are assisted in solving new problems. The environment directs the learner toward the steps of the AMBRE cycle, inspired by the CBR cycle (see Figure 1), as follows:

• The learner reformulates the problem in order to identify problem-structure features (elaboration step of the CBR cycle).
• Then, he/she chooses a typical problem (retrieval step of the CBR cycle).
• Next, he/she adapts the typical problem solution to the problem to be solved (adaptation step of the CBR cycle).
• Finally, he/she classifies the new problem (storing step of the CBR cycle).

Although the system guides the learner by proposing the CBR steps, they are done by the learner him/herself. Contrary to the CBR cycle, the revision step is not included in the AMBRE cycle. Instead, a diagnosis of the learner’s answers is realized on each step of the cycle: a knowledge-based system diagnoses the learner’s answers and, if necessary, provides explanations to help the learner understand his or her mistakes.

We relied on theoretical studies of analogical problem solving in order to specify the processes that learners might carry out on each step of the AMBRE cycle. For instance, during the analysis of a typical problem, they might execute an explanation-based generalization process. In the choice of a typical problem and classification steps, they might carry out processes to detect similarities. During the adaptation step, they might carry out a generalization by adaptation process. Then, based on cognitive psychology studies, we made various
recommendations aimed at facilitating the implementation of these processes in the AMBRE-add ITS. The recommendations are related to the content of the instructions, the way of presenting typical problems, the surface features of the typical problems and problems to be solved, and the type of diagnosis.

**AMBRE-ADD ITS**

We applied the AMBRE principle and these recommendations to the domain of arithmetic. AMBRE-add was designed by a multidisciplinary team of researchers including computer scientists, one cognitive psychologist, and teachers. As outlined above, the objective of AMBRE-add is to teach arithmetic problem classes and techniques associated with each class. According to the AMBRE principle, the system first presents typical solved problems and then guides the learner through the solution of new problems following the steps of case-based reasoning (the AMBRE cycle).

The first phase of AMBRE-add is dedicated to the presentation of typical solved problems (see Figure 2), which are subsequently used as reference problems. During each session, the learner has to read and analyze three typical problems from three different problem classes. A sequential presentation of the solution steps was chosen so as to encourage the student to anticipate the
solution. In order to facilitate comparisons between the typical problems, they all have the same surface features (they are all marbles problems). Moreover, the system presents a report, on the same screen, of all the typical problems analyzed by that learner since he or she first used the system. After several sessions, the learner will have seen a typical problem that matches each problem class defined in the classification.

FIGURE 2
Presentation of a typical solved problem.

In the second phase, the learner has to solve several problems belonging to the various classes of typical problems seen earlier. In order to facilitate the adaptation, the surface features of the first problem to be solved in each class are close to the typical problem’s surface features. The subsequent problems in that class have different surface features.

For each problem, the learner is guided by the system through the five steps of the AMBRE cycle (see Figure 1). In the first step, the problem statement is
provided. In the second step, the learner has to reformulate the problem by identifying the elements in the problem statement that are relevant to the solution (see Figure 3). In order to facilitate the reformulation step, the problems are depicted by diagrams, which have been proposed and tested in didactics of mathematics studies. During the reformulation step, learners have to represent the problem statement using one of the three diagrams. This involves determining what they are looking for (the position of the unknown element) and specifying what they know (the numerical data in the problem). The reformulated problem becomes a reference for the rest of the solving process.

![Image of Amber: Problem Solving](https://example.com/amber_problem_solving.png)

**FIGURE 3**
Reformulation step.

The learner then has to choose the typical problem that is closest to the problem to be solved. Reminders of the typical problems the learner has already seen (at the beginning of the session and during previous sessions) are provided. Also, the initial problem statements and their reformulated formats are given. In
this step, the learner can compare the problem to be solved with typical problems, choose the closest typical problem, and thus implicitly identify the class of the problem to be solved.

The next step is the adaptation of the typical problem. In this step, the learner has to write the solution to the problem while drawing inspiration from the solution in the typical problem he or she chose in the previous step. It is assumed that, through generalization, this adaptation step will lead the learner to identify a problem-solving technique that is suitable for this class of problems. The final step is the classification of the problem. In this step, the learner is presented with a report of the problem’s solving process: the problem statement, its reformulation, and the solution. He or she is then asked to put the problem in the same category as one of the typical problems in order to establish groups of problems, each group being represented by a typical problem. As in the typical problem-choosing step, the learner should compare the problem to be solved with the typical problems.

EVALUATION OF AMBRE-ADD

In order to evaluate AMBRE-add’s usability and its impact on learning, we conducted a series of experiments combining traditional system-evaluation techniques developed in the fields of human-computer interfaces, comparative methods, and qualitative methods developed for the human sciences (e.g. observations, interviews, questionnaires) (see Mark & Greer 1993 for a review).

We first conducted an evaluation with five eight-year-old children who were observed individually in laboratory. The purpose of this experiment was to identify the difficulties encountered by the children and to examine its usability. The results prompted us to apply modifications to the software.

The second experiment, aimed at evaluating the impact of the AMBRE cycle on learning, was conducted in a realistic situation, at a school with three classes of eight-year-old pupils. Pupils that used AMBRE-add were compared to a control group of pupils using a simple problem solving system. The results indicated an improvement in problem-solving performance but no significant difference between the two systems. Observations of the use of the system showed that pupils using AMBRE-add had difficulties reading, understanding messages and navigating. Moreover, the actual use of AMBRE-add was quite different from the intended use. From these observations, AMBRE-add appeared to be too complex for eight-year-old pupils. However, an improvement in problem-solving performance was observed, and the questionnaires showed that AMBRE-add was appreciated both by the pupils and teachers.
We then thought that AMBRE-add would be more suited to nine-year-old pupils. At the age of nine, pupils are better readers and have a greater mastery of arithmetic operations than eight-year-olds, but they still have difficulties modeling arithmetic problems. The purpose of this experiment was twofold: to determine whether the pupils actually used the software in the recommended way and evaluate the effect of AMBRE-add on problem-solving performance. We chose to test the impact of the software itself rather than that of the learning strategy. In this experiment, pupils using the AMBRE-add ITS were compared with pupils who performed the same problem-solving task but did not use a computer between tests; they worked on another activity with the teacher.

The comparison between the two groups was marginally significant (chi²(1)=3.44, p=.064). Most of the children progressed after using AMBRE-add. The few pupils that did not progress solved at least five of the six problems on the pretest so were already high achievers. The observations showed that the nine-year-old pupils were much more autonomous (after the first session, they asked very few questions), understood quickly how to use the system, and did not experience difficulty in reading the help and error messages. Moreover, a substantial number of pupils seemed to use the software as we intended (particularly during the adaptation step). Sixteen of the twenty-one pupils said they used the typical problem to find the solution to the problem to be solved; some of them said they used it when they had trouble solving the problem. These results are encouraging and allow us to consider AMBRE-add as suited to nine-year-olds.

CONCLUSION

In this paper, we described how our multi-disciplinary team designed the AMBRE-add ITS, based on didactics of mathematics studies concerning the domain of arithmetic problems and based on cognitive psychology studies on analogical problem solving. AMBRE-add was designed to teach problem classes and associated techniques based on analogical reasoning by proposing an approach to the learner inspired by case-based reasoning. Lastly, we described how we conducted several experiments in order to evaluate this environment. These experiments were really time-consuming and we should have conducted a first experiment in real situation to analyse learners’ activities on the system before to try to evaluate the impact of the ITS on learning. Therefore, we are going to propose a general approach of evaluation to be implemented during iterative design of ITS.
REFERENCES


