



**HAL**  
open science

## **AMBRE-teacher: a module helping teachers to generate problems**

Stéphanie Jean-Daubias, Nathalie Guin

### ► **To cite this version:**

Stéphanie Jean-Daubias, Nathalie Guin. AMBRE-teacher: a module helping teachers to generate problems. 2nd Workshop on Question Generation during the 14th International Conference on Artificial Intelligence in Education (AIED'09), Jul 2009, Brighton, United Kingdom. pp.43-47. hal-01437698

**HAL Id: hal-01437698**

**<https://hal.science/hal-01437698v1>**

Submitted on 11 Jun 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# **AMBRE-teacher: a module helping teachers to generate problems**

Stéphanie Jean-Daubias, Nathalie Guin  
Université de Lyon, France  
Université Lyon 1 – LIRIS, CNRS, UMR5205  
43 bd du 11 novembre 1918, 69622 Villeurbanne Cedex, France  
{Stephanie.Jean-Daubias, Nathalie.Guin}@liris.univ-lyon1.fr

**Abstract.** If teachers use few ILEs (Interactive Learning Environments) with their students, it may be because they don't have the opportunity to act upon the available environments. To permit the adoption of ILEs by teachers, these systems must be adaptable to the learning context and to the teacher's pedagogical approach. To achieve this we propose a module dedicated to the teacher in the context of the AMBRE project. This module, AMBRE-teacher, allows the user to configure the ILE he wants to use. We identified the functionalities necessary to adapt an AMBRE ILE. We notably designed a knowledge based system allowing the teacher to generate the problems he wants to propose to his students in the ILE. Our approach is implemented for AMBRE-add, an ILE proposing word additive problems in elementary school.

**Keywords.** adaptation of ILE, role of teacher, problem generation, knowledge.

## **1. Introduction**

The AMBRE project is a multidisciplinary study in computer science, cognitive sciences, mathematical didactics and educational science, which aims at designing learning environments for the acquisition of a specific method in problem solving. In each application domain, a method is based on the classification by the learner of problems and solving tools. We propose, to help the learner acquire such methods, to use Case-Based Reasoning, a paradigm developed in Artificial Intelligence and inspired by research in psychology on reasoning by analogy. After the evaluation of a first prototype for the numbering problems domain (final scientific year level, 18 year-old students), we implemented and tested a complete system for additive word problems solving which are studied in primary school: AMBRE-add [7].

In order to facilitate the integration of AMBRE in education, we conceived a module dedicated to teachers, AMBRE-teacher. In this paper<sup>1</sup>, after presenting the functionalities of this module, we describe the functionality dedicated to problem generation and present the architecture of the knowledge based system allowing to implement it.

---

<sup>1</sup> A long version of this paper is available in the research report RR-LIRIS-2009-017.

## 2. AMBRE: “which role for the teacher?”

Most ILEs (Interactive Learning Environments) focus on the computer/learner couple, thus often masking the important role of teachers in these environments [9]. The teacher can first directly participate at the design of the ILE as designer or design partner. He can also be an ILE designer as user of an authoring tool. In addition, the teacher usually takes the role of prescriber by choosing the system that he will propose to his students. The teacher is also sometimes a secondary user of the ILE used by his students when he has to tune it. Lastly and more rarely, the teacher is the main user of a system specifically designed to help him in his teaching task.

In the AMBRE project, if several teachers took part as **design partners** in the framework of differentiated design [5], teachers’ major role is **main user** of a module dedicated exclusively to them. Thus they explicitly take their role of **secondary user** of the AMBRE learner module through the proposed environment, by adapting and defining the parameters of the learner environment. Finally, teachers keep their role of **prescribers**, by choosing the ILE used in their classroom.

For AMBRE, we wish to propose to the teacher an environment designed for him and allowing him to integrate and adapt an AMBRE ILE to his approach, his pedagogical strategies and also to the context of learning. To achieve this, AMBRE-teacher comprises five functionalities. The **configuration of the learner environment** consists in the customization by the teacher of the learner environment interface and of the elaboration of students lists. Even if the learner environment includes a set of predefined exercises, AMBRE-teacher contains a module for **problem generation** enabling teachers to propose to their students problems for which they define the characteristics. Teachers can also **create learning sequences** (sets of problems to solve in one or several sessions of software use) by using the problems they created and by tuning the functioning of the learning environment in terms of help and of diagnosis for the exercises of the sequence. AMBRE-teacher also allows to **distribute the work** to the learners, that is to say to associate the whole group or each learner with one or several sequences. Finally, to allow teachers to entirely adapt the generated problems to the context, AMBRE-teacher can **manage surface features** to be used in the exercises: theme, objects and associated characters, actions, etc.

## 3. A problem generator for AMBRE

We have chosen a semi-automatic generators approach [4] [8] [3], which builds the wording of the problems, but let the user intervene in the creation process. Actually, on the one hand, automatic generators [1] [6], permitting no interaction with users, are not suitable to our aim. On the other hand, manual generators [2], like authoring tools, are not able to solve the problems they allow to create nor to propose any diagnosis of the learner’s answers or help functionalities.

An AMBRE ILE is based on a knowledge based system which relies on a problem solver and allows to provide the learner with help, a diagnosis of his answers and explanations concerning his errors [3]. The problems proposed to the learner must be understandable by the solver to allow the ILE to provide these functionalities.

With AMBRE-teacher, teachers can influence the problems to be generated, by specifying a set of constraints on the exercises to generate. As the problems are built by the system from these constraints, the result of the generation will not only be a wording in natural language, but also a model of the problem usable by the solver.

### *3.1. The problem generation environment for the teacher*

We designed and implemented a tool for problem generation dedicated to teachers, for the word additive problems domain suited to AMBRE-add used in primary school. Problems of this domain describe concrete situations, for example a marbles play: “Alex had 32 marbles. At the end of play, he has 45. How many marbles did he win?”. This domain has been widely studied in mathematical didactics and several problems classifications have been established. The one used in AMBRE-add is presented in [3].

For word additive problems, constraints are of four types. The **structure** of a problem to be generated corresponds to its class, defined by several attributes that can be set or not. **Surface features** are the elements that complete the produced wording. The teacher can specify some elements of this category, for example themes, objects and characters. He can also choose the **values** of the data that will be used in the problems or define an interval for each required values and the wanted difference between min and max values, allowing the carrying over or not, etc. **Complication** concerns all options proposing to complicate the wording of the problem to adapt it to the students’ level. Designing this part required a close collaboration with teachers to identify their needs. The environment proposes language complications and complications of the wording itself. For word additive problems, complication takes the form of vocabulary used and turn of phrases complexity, writing of numbers in full, modification of the sentences order, addition of distractor sentences, addition of non pertinent data. Not all constraints are mandatory for the exercises creation. Constraints not specified by the teacher will be randomly defined by the system.

The four categories defined for word additive problems are not reusable as it for another application domain of AMBRE. Nevertheless, structure features, surface features, and probably complication will still be necessary. Values will only be present for numerical domains.

### *3.2. The GenAMBRE architecture*

The problem generation process that we established in the GENAMBRE architecture takes as input the set of constraints specified by the teacher and gives as output two elements: the wording of the problem in natural language for the learner and a computer-usable formulation of the wording named descriptive model of the problem, for AMBRE problems solver.

The problem generator architecture for AMBRE is presented in **Figure 1** and each of its components is presented in the following of the section. For a D domain (for example the additive word problems domain), the five knowledge bases of the *domain level* required by the *generation level* must be defined by using the domain independent formalisms of knowledge representation. The problem generation process is done in

two stages: the system builds a problem generation model, then builds the wording in natural language and the descriptive model of the problem. Both these processes are domain independent. Both processes and the knowledge bases of the D domain, constitute together a problem generator for the D domain: GenAMBRE-D.

**Classification knowledge** For each application domain, an expert gives to AMBRE solver, and consequently to GenAMBRE generator, a problems classification graph. This hierarchy of classes is used by the solver to classify the problem. This is a domain dependant class hierarchy, but its representation is the same for all domains.

**Knowledge of the themes** To generate a problem, it is necessary to know the concerned theme and the associated surface features (for example objects, characters and actions). Knowledge of the themes is given by the expert, or created by the teacher himself, through the surface features management module of AMBRE-teacher.

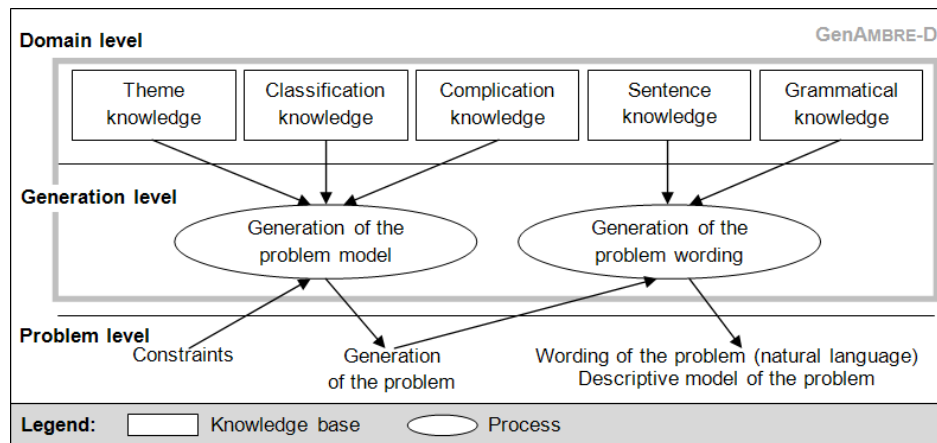


Figure 1. The GenAMBRE architecture.

**Complication knowledge** For additive problems, complicating a wording mainly consists in changing the sentences order and adding distractor sentences. So complication knowledge answers the following questions: how can one modify the order of the sentences of the problem? What distractor sentences can we add to problems and where can we place them?

**Generation of the problem model process** From the three knowledge bases previously described and from the constraints keyed in by the teacher, the system has to generate what we call a generation of the problem model. This model is an extensive descriptive model, because it also details the problem class and its theme. To build this model, the process fulfills the constraints defined by the teacher, notably by choosing random values for the undefined constraints.

**Grammatical knowledge** The domain expert must furnish a grammar to the generation system: a set of sentences structures that could be used in the domain.

**Knowledge on the sentences** The generated sentences, notably their structure, depend on the class of the problem. It is therefore necessary to know what sentences structures (from the grammar) could be used for the problem to be generated to permit

to generate the wording in natural language. So, knowledge on the sentences allows to associate the class of the problem to the usable sentences structures, and the associated elements of the problem.

**Generation of the problem wording process** To generate a wording in natural language, the process uses knowledge on the sentences and domain grammar, as well as the generation of the problem model previously created. Knowledge on the sentences allows the system to take conceptual decisions (deciding what to tell), then the process goes to the text generation step and establishes syntactical treatments (deciding how to tell it) and lexical and morphological ones (deciding how to write it).

#### 4. Conclusion

In this paper, we presented how we designed a module dedicated to teachers for the AMBRE-add ILE, to allow them to adapt the ILE to the learning context and to their pedagogical approach. By adopting a generic approach, we identified, with the help from teachers, functionalities that a teacher module must propose for an AMBRE ILE (AMBRE-teacher). We have also enabled the teacher to configure the environment, generating problems suited to his needs, creating learning sequences suited to his students by choosing the problems and the behavior of the ILE, assigning these sequences to his students, and creating new themes of exercises.

These functionalities are implemented for the word additive problems domain. For this, we designed a problem generation system whose architecture is domain independent. Even if we integrated teachers in the design of AMBRE-teacher, it is now necessary to evaluate it in actual classroom situations with a significant number of teachers. Finally, we must also validate the functionalities defined for AMBRE-teacher and the genericity of the GenAMBRE architecture by implementing a teacher module for an AMBRE ILE for another domain.

#### 5. References

- [1] Burton, R. R. (1982). Diagnosing bugs in a simple procedural skill. *Intelligent Tutoring Systems*. London, Academic Press, 157-184.
- [2] David, J.-P., Cogne, A. and Dutel, A. (1996). Hypermedia exercises prototyping and modelising. *Computer Aided Learning and Instruction in Science and Engineering*. S. B. Heidelberg, 252-260.
- [3] Duclosson, N. (2004). Représentation des connaissances dans l'EIAH AMBRE-add, TICE'2004 conference, France, 164-171.
- [4] Giroire, H. (1989). Un système à base de connaissances pour la génération d'exercices dans des domaines liés au monde réel, PhD Thesis, Université Paris 6.
- [5] Jean-Daubias, S. (2009). Differentiated design: a design method for ILE. Research report RR-LIRIS-2009-015.
- [6] Mitrovic, A., Stoinemov, L., Djordjevic-Kajan, S. (1996). INSTRUCT: Modelling Students by asking questions, *User Modelling and User-Adapted Interaction*, vol.6-4, 273-301.
- [7] Nogry, S., Jean-Daubias, S., Duclosson, N. (2004). ITS Evaluation in Classroom: The Case of AMBRE-AWP, ITS'2004 conference, 511-520.
- [8] Pecego, G. (1998). SYGEP, un Système de Génération d'Énoncés de Problèmes dans des domaines variés, PhD Thesis, Université Paris 6.
- [9] Vivet, M. (1990). Uses of ITS: Which role for the teacher?, *New Directions for Intelligent Tutoring Systems*, NATO ASI series, Vol. F91, Springer-verlag, Sintra.