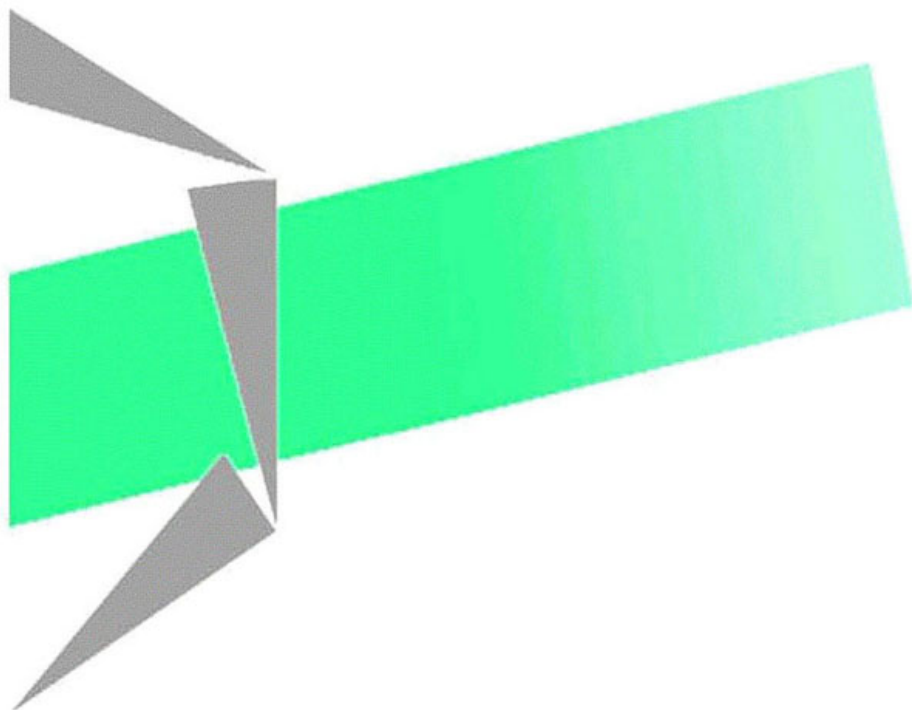


Les cahiers Leibniz



Developing Interactive Learning Environments that can be
used by all the classes having access to computers.
The case of Aplusix for algebra

Jean-François Nicaud, Denis Bouhineau, Hamid Chaachoua,
Jana Trgalova

Laboratoire Leibniz-IMAG, 46 av. Félix Viallet, 38000 GRENOBLE, France -
ISSN : 1298-020X

n° 148
Mar. 2006

Site internet : <http://www-leibniz.imag.fr>

Developing Interactive Learning Environments that can be used by all the classes having access to computers. The case of Aplusix for algebra.

Jean-François Nicaud¹, Denis Bouhineau¹, Hamid Chaachoua², Jana Trgalova³

¹ Université Joseph Fourier, ² IUFM de Grenoble, ³ IUFM de Lyon

Leibniz-MeTAH, 46 Ave Viallet, 38031 Grenoble, France

{Jean-Francois.Nicaud, Denis.Bouhineau, Hamid.Chaachoua, Jana.Trgalova}@imag.fr

Abstract. Our research team has developed and experimented software for the learning of algebra, named Aplusix, with the idea of being usable and useful for all the classes having access to computers, and of helping teachers to teach the curriculum. In this paper, we list 19 principles that we consider relevant to this goal and we briefly describe the Aplusix system. This system is distributed in France since early 2005 and will be distributed in many countries from 2006. It has proven to be efficient (students learn) and to facilitate the teacher's work.

Themes.

Learning and assessing mathematics with and through technologies

Issues for design of learning environments and curricula.

1. Introduction

Interactive Learning Environments (ILEs) is a term used in the AI-ED community (Artificial Intelligence in Education) for computer systems designed to help students learn a domain in an interactive way¹. ILEs can be produced by research teams or companies. Research teams generally develop prototypes² in order to implement and experiment ideas; generally, they do not develop products³ that can be used in ordinary classes by teachers who just want to teach the curriculum. Companies develop products to be used in ordinary classes. Most of these products do not include advanced functionalities and have a poor interactivity, which is sometimes limited to multiple choice questions.

Many mathematics teachers complain they do not have enough time to teach the curriculum. This produces a strong limitation to the use of ILEs because these

¹ ILEs do not include all the pieces of software that can be used at school, for example CASs and spreadsheets are not ILEs, they are professional software. The fact that they can be used in classes, at some level, in well-prepared situations, does not make them ILEs.

² A prototype is an application which is not well finished and debugged and cannot be distributed as a professional system.

³ A product is a well finished and debugged application which can be distributed as a professional system.

systems consume a lot of teachers time (study of the system, preparation of learning situations) and time of their students (it replaces well known activities by activities in the computer room having an efficiency which is not obvious).

This paper is devoted to ILEs in mathematics (more particularly in algebra) to be used in the usual framework of classes that can access to computers, both in developed and developing countries. We particularly focus on ILEs that can help students learn the curriculum with little modification of the class functioning and without the goal of changing what is learnt. We consider that this is the first kind of ILE that most of the teachers in the world need. However, we do not mean that this is the only interesting kind of ILE.

In section 2, we list principles of ILEs *for the usual framework of the class* and a few algebra ILEs are situated with regards to these principles. In section 3, we present the Aplusix ILE project in algebra. After a first period where prototypes were built and experimented, we adopted the goal of building an algebra ILE *for the usual framework of the class* and we redesigned our system according to this goal. At the present time, experiments have shown that the system verifies these principles. We are now entering in a deployment phase.

2. Principles of ILEs for the usual framework of the class

The principles we consider are oriented towards the students or the teacher.

1. The tasks proposed by the ILE must be part of the curriculum.
2. The activities must be close to the usual activities of the curriculum. This includes a place for errors and a general interaction mode with little scaffolding (there is scaffolding when the system executes a part of the task).
3. The main representations at the interface must be close to the usual ones. Other representations must be added only for didactical reasons.
4. There must be transfer on paper. After an adequate amount of activities with the computer, students must have better scores on paper tests.
5. Teacher must be able to use the ILE in the theoretical framework, either explicit or implicit, they are used to.
6. The domain of the ILE must be large.
7. The manipulation of the representations must be natural (not involving intermediary representations) and easy.
8. The ILE must be in the natural language used in the school.
9. The familiarization with the system must be easy. When the overall interaction is complex, there must be different stages so that the familiarization of each stage is easy.
10. When important features of the system depend on human choice, parameters must allow teacher's decision on the behavior of the system.

11. The ILE must present some added value compared to traditional environment.
12. The ratio between the time for familiarization and the duration of learning activities must be low (maximum 20%).
13. The interaction modes and the feedbacks must allow a good level of autonomy of the students when they use the ILE, so that the load for the teachers is not heavy.
14. The time the teachers need for preparing learning situations must be short.
15. Information concerning the student activities must be accessible to the teachers.
16. The teachers must be involved in the global learning process.
17. The price of the system must be adapted to schools. When the system requires the use of another system, like a CAS, this system must not be expensive, and must be easy to obtain or included in the installation package.
18. The installation of the system must not be complex, because it is often done by teachers who have no advanced knowledge in computer science.
19. The organization developing the ILE must be durable (10 years or more), because an ILE must evolve (correction of bugs, development of new functionalities, etc.).

Principles 1 to 6 are *adequacy* and *utility* principles. Principles 7 to 10 are *usability*. Principles 11 to 14 are *economical principles* at a cognitive level. Principles 15 and 16 exclude AI systems that would take in charge the entire learning process and leave no place to the teacher. Principles 17 to 19 are general *economical principles*.

Cognitive tutor for algebra [Koedinger et al., 1997] is an example of ILE *for the usual framework of the class*. Its domain is elementary algebra: word problems, linear equations and systems. It has been designed by the Carnegie Mellon University of Pittsburgh and is distributed by the *Carnegie Learning* company [CarnegieLearning], which is a spin-off of this University. It is oriented to the US curriculum. The Website of *Carnegie Learning* announces “325 000 students using the system over 750 school districts across the United States” and better scores of students using this system with regard to other students.

MathXpert [Beeson, 1996] is an ILE for algebra and calculus designed at the University of San José and distributed by the *MathWare* company [MathWare]. Experiments have proven that students benefit from the use of MathXpert. However, MathXpert does not follow the principle 2: it provides strong scaffolding (at any time, the student selects a sub-expression and the system provides a menu with the rules that can be applied to this sub-expression) and makes no place for errors (the student chooses a rule in the menu and the system applies this rule). This non respect of the principle 2 may be a reason of a weak success of the system.

3. The Aplusix ILE

3.1. Brief history of Aplusix

During the nineties, we developed an ILE for algebra [Nicaud et al., 1990]. It was a command-based ILE like MathXpert with rules in menus and calculations done by the system. The main differences with MathXpert were: (1) more abstract rules; (2) Presentation of all the rules (not only the applicable ones); (3) Demand of the values of the rules variables; (4) A small domain (factorization of polynomials). Several experiments were conducted and provided good results.

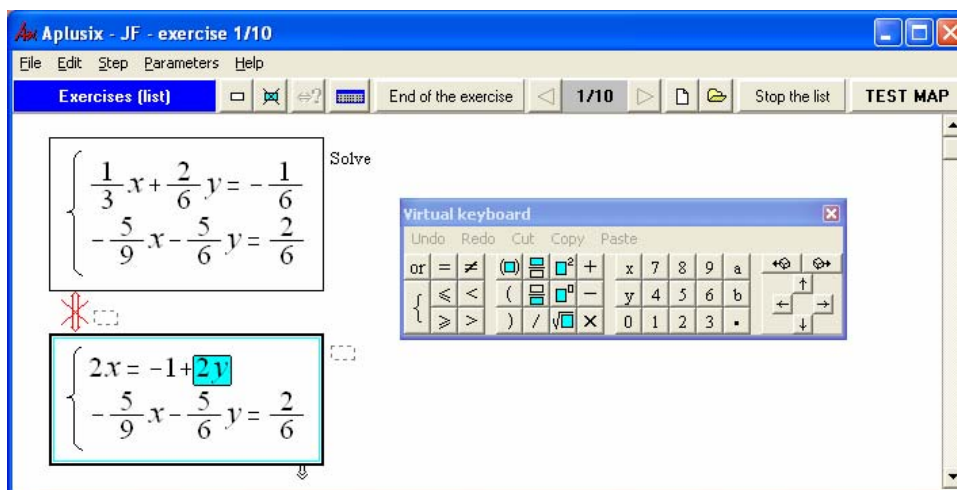


Figure 1. The Aplusix ILE.

As researchers in computer science and developers of an ILE, we were not satisfied with the usage of the system: it was used only in small experiments driven by researchers. So, we decided to redesign the systems with the goal of being attractive and usable by many teachers and students. At this moment, we did not have the *ILE for the usual framework of the class* idea but a will to have a system that can be widely used like dynamic geometry software [Laborde, 1989]. So we built the Aplusix system [Nicaud et al., 2004], see figure 1, with the following main features: (1) To allow the student to freely build and transform algebraic expressions and solve algebra exercises by producing his/her own steps like on paper; (2) To produce the *first fundamental feedback*, the indication of the correctness of the steps, in a non intrusive way, see figure 2. This follows principles 2, 3, and 7. According to the principle 17, we decided to develop ourselves the entire system (without using a CAS or some other piece of software). For feature (1), we developed an advanced editor of algebraic expressions and for feature (2), we developed a module of formal calculations including the calculations of the equivalence of expressions.

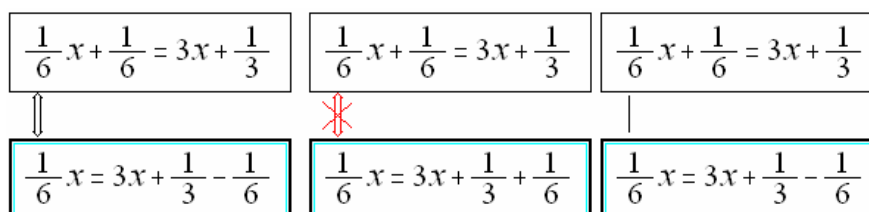


Figure 2. A correct step in the left, with an equivalence sign; an incorrect step in the middle, with a red crossed equivalence sign; a step without feedback, in the right.

The teachers and the students who used the first version of this new Aplusix were very interested and the first tests showed that the students learned well. However, the teachers had soon demands of new functionalities: (1) A *second fundamental feedback*, the indication of the correct end of the exercises; (2) A mode without feedback; (3) Exercises ready to be used. At this moment, we entered in the *ILE for the usual framework of the class* idea.

3.2. Short description of Aplusix

First versions of the new Aplusix were principally pieces of software devoted to students, based on the concept of microworld, with a rich replay system usable by students, teachers and researchers. In last versions, we continued to improve the microworld aspect of our environment, but we worked to nest it into some kind of exerciser where the work done by students could be automatically analyzed and scored, *fundamental feedbacks* can be hidden to permit use of Aplusix for tests, exercises could be automatically generated, solutions could be automatically found out. Both students and teacher were targeted by these new components, and their works have been facilitated. But we have worked for teachers specifically too, trying to provide them with tools for the administration of their classes and of their students, tools for the edition of specific exercises, list of exercises, or richer exercises (for word problem, or problems with many linked sections), and last, tools for statistical analysis of student's results, see figure 3.



Figure 3. Statistical analysis of student's results. First the total of well-solved exercises, second, the total of attempted exercises.

As a consequence, the main student's activity envisaged with Aplusix is no more an exploration activity with the microworld but a training activity on a list of given exercises under the control of *fundamental feedbacks* or a test activity when these feedbacks are hidden, these new activities are closer to the one practiced in class. We have added also a new activity, we call self-correction, where students, after a test (i.e., without *fundamental feedbacks*), can benefit from *fundamental feedbacks* to correct their errors. The last activities are visualization of past activities, either globally (the final form of the exercise) or action by action.

The organization of work with Aplusix according to activities has been a solution to reduce the use of the parameters. A set of parameters allows customizing the system for each class and situation. Parameters continue to exist and can define for examples: the mode and scope of the verification of the equivalence; how the system

must manage an incorrect or ill-formed step; the access to the solutions and to the CAS-like commands, the order of exercises obtained from a list (randomized or not); the introduction of strong invitation to students to comment their steps. See figure 4. Activities do not set all the parameters but the most important ones and reduce the number of those which still need to be set. Because of the complexity of the use of parameters (set of parameters can be assigned to each class, for each session), there was a real need to find a way to have customized version of the system without big effort. Activities have been our solution to make the system easier to use.

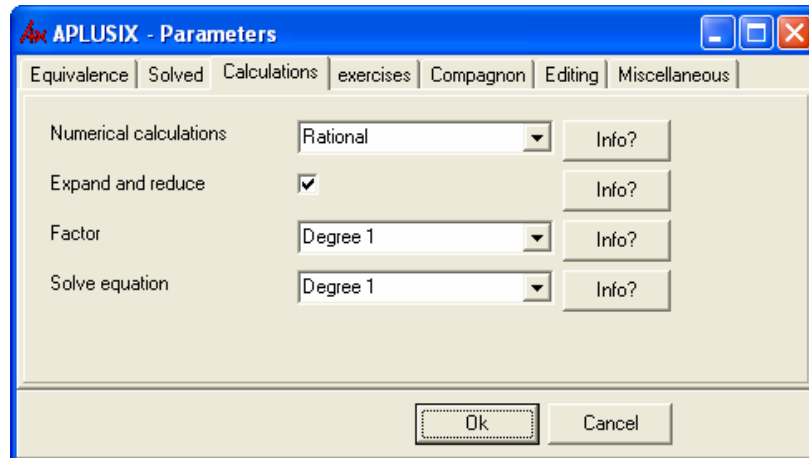


Figure 4. Teacher's panel for choosing the values of the parameters.

The domain available for explorations and exercises concern algebra: numerical calculations (from integers to square roots), expansion of polynomial expressions of several variables, factorizations of polynomial expressions of one variable and maximum degree 4, polynomial equations and inequations of one unknown and maximum degree 4, rational equation leading to polynomial equations of one unknown and maximum degree 4, system of linear equations up to 10 equations and 10 unknowns.

For more precise information consult documentations at <http://aplusix.imag.fr>

3.3. Current state of the global project

Aplusix has been developed in French. We have developed tools allowing to easily translate the texts and the help file of the system. Aplusix has now been translated in English, Portuguese, Italian, Vietnamese, and Arabic. Translations are ongoing in Spanish and Japanese.

Until the end of 2004, several experiments have been carried out [Nicaud, 2005a] in different countries (France, Brazil, Italy, Vietnam, and India) and contexts (a few sessions, regular use during the entire school year, one student per computer, two or four students per computer) for a total of about 15 000 students*hours. At the present time, principles 1 to 18 are verified. In particular: (1) students gain autonomy and improve their knowledge; (2) Aplusix facilitates the teachers' work (because of students' autonomy and of already-made lists of exercises). Furthermore, enquiries showed that all the students worked with pleasure with Aplusix.

Early 2005, a contract was signed between our University and a French publisher, and Aplusix began to be marketed in France. This type of contract appears to be unsatisfactory with regards to duration (principle 19), because, according to French law, the royalties received by the University cannot be used to pay engineers for maintaining and developing the system. So we decided to move the development structure. First, it will go for 4 years to a company, which is an affiliate of the University; then a spin-off company will be created. We have contacts, for several months, with publishers who are willing to sell Aplusix out of France. We will be able to sign contracts with them in March or April 2006. We will adapt the price to the gross domestic product of the countries.

4. Discussion

The above list of principles has been built from our view of ILEs *for the usual framework of the class*. As Aplusix has this goal for several years, it is not by chance if it follows these principles. However, we may have forgotten some principles in the elaboration of this list and ideas are welcome.

This list may also be used to estimate the distance between an ILE (an existing ILE or an ILE to be developed) and the *for the usual framework of the class* concept. An important distance does not mean a poor ILE (the ILE may be very good), but we think that an important distance means a limited possible use. In that case, the benefit of the development of the ILE is only at a research level (doing experiments, producing results of these experiments, publishing papers), not at a usage level.

In the case of Aplusix, we are developing new functionalities that will help students and teachers (through students' autonomy): (1) a companion as an ideal student of a given level who can provide suggestions, explanations and calculation steps; (2) a tutored mode in which the students' calculation steps will be analysed and when a misconception is diagnosed [Nicaud at al. 2005b], an adequate feedback is provided, for example: *When you move an additive expression from one side to the other side of an equation, you must change its sign.*

5. References

Beeson M. (1996). Design Principles of Mathpert: Software to support education in algebra and calculus, in: Kajler, N. (ed.) *Human Interfaces to Symbolic Computation*, Springer-Verlag.

CarnegieLearning <http://www.carnegielearning.com/>

Koedinger, K. R., Anderson, J. R., Hadley, W. H., Mark, M. A. (1997). Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education*, 8, p. 30-43.

Laborde J.M. (1989). Designing Intelligent Tutorial Systems: the case of geometry and Cabri-géomètre, *IFIP WG 3.1 Working Conference on Educational Software at the Secondary Education Level*, Reykjavik.

MathWare <http://www.mathware.com/mathxpert.html>

Nicaud J.F., Aubertin C., Nguyen-Xuan A., Saïdi M., Wach P. (1990). APLUSIX: a learning environment for student acquisition of strategic knowledge in algebra. Proceedings of PRICAI'90. Nagoya.

Nicaud, J.F., Bouhineau, D., and Chaachoua H. (2004). Mixing microworld and CAS features in building computer systems that help students to learn algebra, in *Int. Journal of Computers for Mathematical Learning*, Volume 9 (2), Springer-Verlag.

Nicaud J.F., Bittar M., Chaachoua H, Inamdar P., Maffei L. (2005a). Experiments of Aplusix in Four Countries. 7th International Conference on Technology in Mathematics Teaching, Bristol.

Nicaud J.F., Chaachoua H., Bittar M., Bouhineau D. (2005b). Student's modelling with a lattice of conceptions in the domain of linear equations and inequations. Workshop proceeding of the AI-ED conference, Amsterdam.

Le laboratoire Leibniz est fortement pluridisciplinaire. Son activité scientifique couvre un large domaine qui comprend aussi bien des thèmes fondamentaux que des thèmes très liés aux applications, aussi bien en mathématiques qu'en informatique.

Les recherches sur les Environnements Informatiques d'Apprentissage Humain et la didactique des mathématiques ouvrent cette pluridisciplinarité sur les sciences humaines, elles jouent un rôle particulier en favorisant les coopérations entre différentes composantes du laboratoire.

- * mathématiques discrètes et recherche opérationnelle
- * logique et mathématique pour l'informatique
- * informatique de la connaissance
- * EIAH et didactique des mathématiques

Les cahiers du laboratoire Leibniz ont pour vocation la diffusion des rapports de recherche, des séminaires ou des projets de publication réalisés par des membres du laboratoire. Au-delà, Les cahiers peuvent accueillir des textes de chercheurs qui ne sont pas membres du laboratoire Leibniz mais qui travaillent sur des thèmes proches et ne disposent pas de tels supports de publication. Dans ce dernier cas, les textes proposés sont l'objet d'une évaluation par deux membres du Comité de Rédaction.

Comité de rédaction : Nicolas Balacheff, Mirta Gordon, Denise Grenier, Yves Demazeau, Andras Sebo

Contact Gestion & Réalisation : Jacky Coutin
Directeur de la publication : Nicolas Balacheff
ISSN : 1298-020X - © laboratoire Leibniz