HAL open science

Helping Teachers Generate Exercises with Random Coefficients

Denis Bouhineau, Hamid Chaachoua, Jean-François Nicaud

To cite this version:

Denis Bouhineau, Hamid Chaachoua, Jean-François Nicaud. Helping Teachers Generate Exercises with Random Coefficients. International Journal of Continuing Engineering Education and Life-Long Learning, 2008, 18 (5/6), pp.520–533. hal-00961967

HAL Id: hal-00961967

https://hal.science/hal-00961967v1

Submitted on 25 Mar 2014

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.

Helping Teachers Generate Exercises with random coefficients

Denis Bouhineau, Hamid Chaachoua & Jean-François Nicaud

University of Grenoble – LIG Laboratory
46 avenue Félix Viallet, F-38031 Grenoble Cedex, France
E-Mail: Denis.Bouhineau@imag.fr
E-Mail: Hamid.Chaachoua@imag.fr
E-Mail: Jean-Francois.Nicaud@imag.fr

Abstract: First, we propose two taxonomies concerning software designed for teaching mathematics, which we call TEL (Technology Enhanced Learning) environments, the first one concerns the teacher's place and role in TEL environments, the second one concerns the activities which are foreseen and provided by TEL environments. Second, we consider TEL environments which provide teachers with tools for building patterns of exercises used in the TEL environment to generate randomly and dynamically exercises or lists of exercises. Our approach is compared to classical approaches (based on standards like IMS-QTI or on Computer Algebra Systems).

Keywords: Design of TEL (Technology Enhanced Learning) environments, Pattern of exercises, Tools for Teachers, Training and Test activities, Pedagogical and Didactical Parameters.

Biographical notes: Denis Bouhineau is associate professor at the Joseph Fourier University in Grenoble, France. He received his PhD degree in computer science from the same university in 1997. He teaches undergraduate and graduate computer science courses. His research focuses on design of TEL environments for active learning situations used in real world.

Hamid Chaachoua is associate professor at the Joseph Fourier University in Grenoble, France. He earned his PhD degree in didactics of mathematics from the same university in 1997. He teaches pre-service teachers in mathematics. His research focuses on Didactics of Mathematics used in TEL environments.

Jean-François Nicaud is professor at the Joseph Fourier University in Grenoble, France. He earned his PhD degree in computer sciences from the university of Paris-Sud in 1987. He teaches undergraduate and graduate computer science courses. His research focuses on design, development and deployment of TEL environments for algebra used in real world and initiated the Aplusix Project in the 80'.

1 Introduction

Since 2000, we are working on the TEL (Technology Enhanced Learning) environment Aplusix dedicated to the teaching and learning of formal algebra at secondary school

Copyright © 200x Inderscience Enterprises Ltd.

level, see figure 1. Our main goals were to develop from scratch a high-quality TEL environment useful for research which would works also really at school : easy to use and efficient for learning/teaching. Principles worth following to achieve these goals and lessons learned from our experience have been published recently in [NIC-06].

During a first phase, from 2000 to 2003, most of our work has been devoted to the design of a microworld for *learning-by-doing* elementary algebra [NIC-04]. Main principles used to develop this environment were, first, to let students freely and intuitively build and transform algebraic expressions, as they can do on paper, and second, to give permanent but not intrusive epistemic and fundamental feedbacks on the syntactic and semantic correction of the algebraic reasoning followed by the students. Experiments and usage in usual classes with this microworld have been successful, they have been reported in [CHA-04] and [BOU-05a]

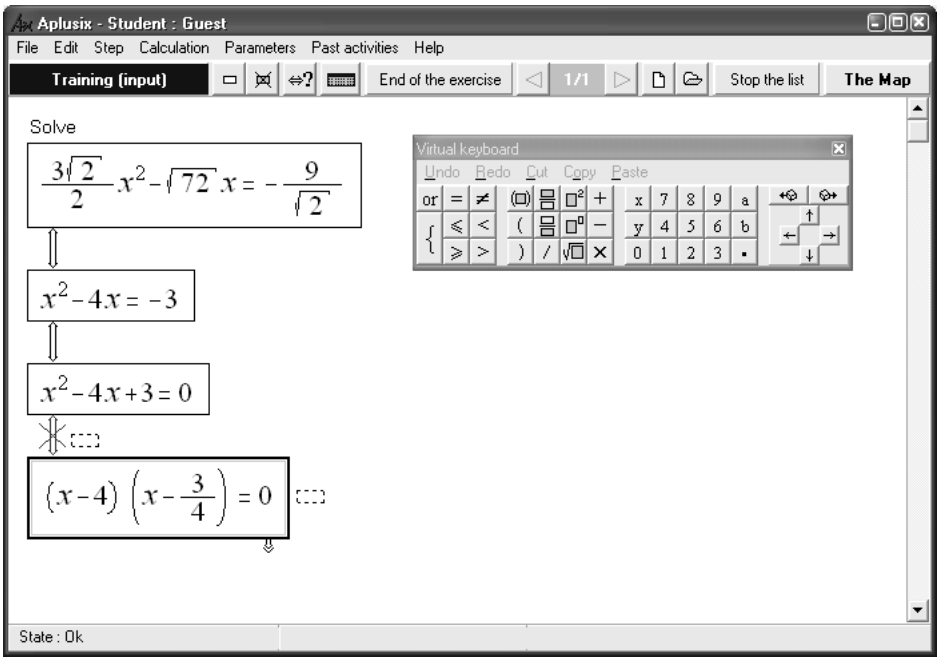

Figure 1 Few steps of a student's reasoning in Aplusix to solve a quadratic equation. The first box contains an equation to be solved by the student. The other boxes have been built by the student thanks to the advanced editor of algebraic expressions of the system. The two first steps are correct: the modified equations have the same solutions as the previous one (1 and -3). The last step is not correct: a cross –red on computer screen- is draw over the double equivalent arrow.

The second phase, from 2003 to 2005, introduced a shift in the kind of TEL environment we were working on: we moved from the microworld paradigm to an exerciser paradigm. We worked specifically for math teachers, trying to provide them with tools for managing, improving and facilitating their work. One of our first works was to include an exercises generator for a wide range of algebraic exercises in all the domains available in Aplusix. Then, we introduced two modes for doing the activities in the software: one for training (with feedback, help and hint), one for tests (without feedbacks, help and hint, and with a limited amount of time). In parallel, we developed

tools for editing richer exercises and for organizing scenarios of use. We also improved the feedback given to teachers for the analysis of the student's activities. And, finally, we facilitated the installation of the software on network.

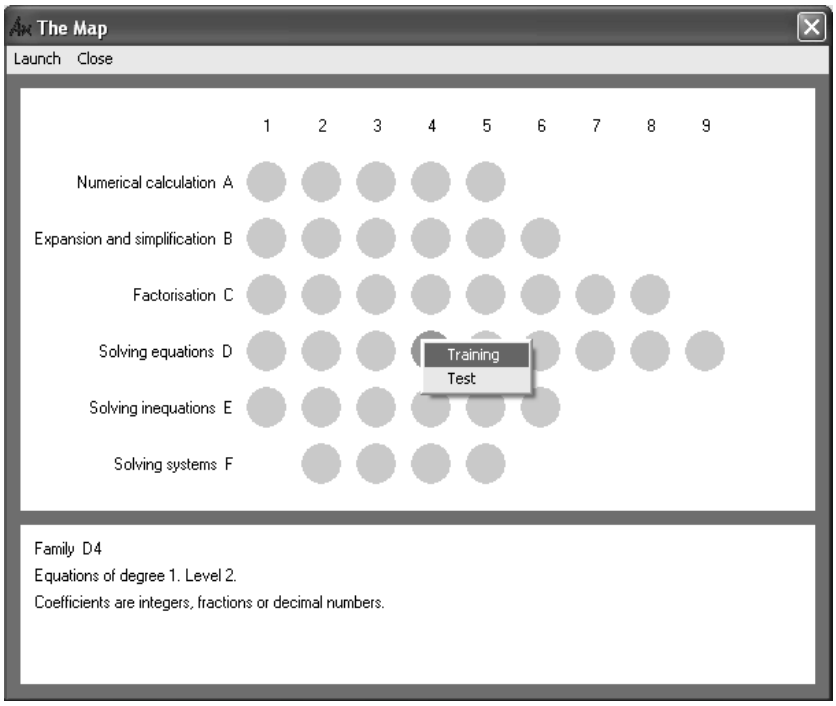

Figure 2 The map of exercises. When the student clicks on a dot, s/he gets a description of the family in the bottom part and a menu for launching the family in the training or test mode.

Thus, we added to Aplusix a map of algebraic exercises, see figure 2. This map of exercises is a set of about 40 families of exercises in formal algebra organized by type of exercise, domain and level of difficulty. The organisation is partly based on didactical variables, partly based on pedagogical reasons. In few words, behind the stage, each family contains about 10 patterns of exercises. More than 500 patterns of exercises have been produced for formal algebra exercises at secondary school level. Not all were used. Each pattern of exercises is defined with a general pattern having random coefficient. For each coefficient a domain of possible values is given. In the pattern, calculi are also described to be performed on the value and on the instantiated exercises. Few elements are also described: mean time to solve the exercise, exercise type, hints to describe the pattern, general family of pattern, pseudo-frequency to set the number of occurrences of the pattern. Finally, and this constitutes an original contribution of that work, the pattern contains constraints to be satisfied: theses constraints are to be tested out to validate an instantiated exercise. These constraints represent a natural way to express that the expected exercises are like the general pattern (with some defined properties given by the general pattern, the calculus and the domain of the coefficient) but not having other properties identified by the constraints. The introduction of such constraints has been

decided to ease the work of math teachers and didacticians willing to participate to the definition of the patterns. As a consequence, the exercises are generated from patterns by non trivial random choices of the numbers and with verification of the constraints; see [BOU-05b] for more information.

The families of exercises and the patterns of exercises used in the map of exercises have been built by didacticians in mathematics and an important work has been made with them to define the language used for patterns. The introduction of the concept of 'constraint to be satisfied' is fruit of that effort and has been validated by the effective construction of the map of exercises. But each pattern remains a complex object with a dozen of attributes, see figure 3. The language used to described patterns has a strict syntax (as almost all the formal languages), and unfortunately no editor have been made available for the definition of the map of algebraic exercises. The definition and realization of an editor is time-consuming and were not in our priorities at that time. As a consequence, the process for the writing of the map of exercises was not straightforward: a first version of the patterns was written by didacticians in mathematics, some work was needed then, made by computer scientists to debug some patterns and assemble the patterns in the map of exercises. Then, tests were made by didacticians to validate the exercises and improve them when necessary. Improvements or additions were then possible, restarting the loop or the spiral of design.

```
{[Name                                          FactorDistSTD23Z]
 [KindOf                                           FactorDistSTD]
 [Hint          "Std factorization with numerical fact., no cst"]
 [TypePb                                           Factorization]
 [MeanTime                                                     3]
 [Frequency                                                    1]
 [Pattern                                            <<ax^2+bx>>]
 [Domain                                       ((c integer+ small)
                                                (d integer+ small)
                                               (e integer* small))]
 [With                                           (       (<> c 1)
                                                   (:= a (* d c))
                                                   (:= b (* e c)))]}
```

Figure 3 Example of a pattern: x remains abstract variable, a and b are instantiated with products d.c and e.c where c and d are small strictly positive integers, e is a non-null, c is non-unitary integer. For example, that pattern can produce the following expression $18x^2+30x$, $12x-9x$ and $14x^2+7x$. Factoring these expressions should reveal the c coefficient. With these examples, it should leads to $6x(x+5)$, $3x(4x-3)$ and $7x(2x+1)$.

Our work for helping math teachers on the automatic generation of random exercises has been the opportunity to make short studies about the teacher's place and role in TEL environments and the activities which are foreseen and provided by TEL environments. The first part of this paper concerns these two taxonomies. Main focus is on TEL environments in mathematics and activities performed in mathematics, but not only.

The second part of this paper is at the cross-road of the two taxonomies, and considers TEL environments which provide math teachers with tools for building patterns of exercises and generating randomly and dynamically lists of exercises. Our approach, and our last developments, will be compared to other typical and standard approaches.

2 Taxonomies

2.1 Teacher's place and role in TEL environments

The first short taxonomy considers teacher's place and role in TEL environments. The goal is to observe particularly if specific parts of TEL environments have been designed for the teacher, and if the answer is yes, what are theses parts. With these questions in mind, and after a short informal study on a set of 21 TEL environments, mostly in the domain of mathematics, four groups can be formed (from the less to the most important place and role offered to teachers):

- **T1. TEL environments that simply ignore teachers.** They are TEL environments in which no part of the environment has been designed for teacher's use specifically. No functionality is devoted to the teacher. Teachers and students have the same environment. It does not necessarily mean that teacher has no role in the use of the TEL environment, because it does not necessarily mean that the TEL environment replace or acts as a teacher.

  Actually, two subgroups may be defined for group T1, the first one (T1.1) where the environment supposes the presence of a teacher who has to organise the activities (create documents on paper, or files to open, conduct and validate him/herself the result of the activity…). In this subgroup, we can place a lot of microworlds. We could also place pieces of software which are not TEL environments but are used in classroom (text editor, calculus sheet…). TEL environments of subgroup T1.1 can be viewed as tools for teachers. Compared to pieces of software which are not TEL environments, they intend to offer a pedagogical added value.

  The second group (T1.2) collects the environments which incorporate some kind of electronic teacher to organise the activities (choice of the exercises, diagnosis of the answer…) Most of ITSs – Intelligent Tutoring Systems – belong to that subgroup. At the opposite of the TEL environments of subgroup T1.1, environments of subgroup T1.2 do not need teachers; teachers are just ignored by these environments. Student can use these environments at home, or autonomously in classroom (this could be interesting for teachers willing to work specifically with some other students, and can let the first one work alone).

- **T2. TEL environments that give feedback to teachers.** Some TEL environments from the previous subgroup T1.2 can give feedback, for example a user model, it consists essentially in showing externally the internal representation of the students, or an historic of the student's actions with simple and local diagnosis of the success or failure of the student. Such feedback can be of importance and interest for the teachers. Some TEL environments have rightly understood that and focus on it. These environments explicitly introduce the role of the teacher as the person for which these feedbacks are given. These TEL environments constitute the subgroup

T2.1.

Even more, some TEL environments have been specifically designed to give feedbacks to teachers (T2.2). An example is given by TEL environments which are dedicated to *competencies diagnosis.* These environments are used at the beginning of the year, or before inscription. Usually, TEL environments from that subgroup (T2.2) are not used in class during normal time and their use is very limited. So that subgroup is very small.

However, in most of the case, when a TEL environment gives feedback to teachers, it does not belong to group T2, but belongs to one of the two last groups T3 or T4: the place and role of the teacher is larger, in that case, the teacher's role is not only to receive feedbacks.

- **T3. TEL environments with teacher's interface to manage classes.** The role and place given to teachers in these environments is limited to managing their classes, students and activities (among, most of the time, only predefined activities). Most of the time, teachers can also obtain feedbacks from the environment about the success and failure occurred in the passed activities. Sometimes, environments from group T3 let the teachers specify the behaviours of the TEL environments with pedagogically and didactically meaningful parameters (presence or not of feedback or hint, time to achieve the activity…).

  The limitation occurs when the teacher wants to modify the activities predefined in the environment. The main difference between T3 and T4 is the possibility to add new exercises and to organise the scenario of his/her own activities.

- **T4. TEL environments with teacher's interface to manage classes and edit activities.** These environments propose teachers to manage their classes, students and activities, to specify the behaviours of the environments and to easily modify or enrich the environments with their own activities (lessons, exercises, tests).

  Two subgroups can be considered in group T4, according to the competencies needed by the teachers to perform all the roles devoted to him. Subgroup T4.1 is for TEL environments asking for no noticeable competencies in computer science. Subgroup T4.2 is for TEL environments asking for noticeable competencies in computer science.

  In subgroup T4.1, TEL environments must include very specific parts and functionalities to permit the introduction or edition of one core part of the TEL environment. It does imply an important supplementary work for the designers.

  In both cases, T4.1 and T4.2, an important supplementary work is asked to the teacher when s/he wants to add new exercises, and the learning curves to control the given tools are to be considered. Is it worth? Could teachers afford it? Clearly, the teacher has to learn how to use that specific part of the TEL environment. It is worth noting that this specific part is very often delicate and complex. Next, the teacher has to effectively introduce or change the activities. It could mean possibly that, compared to the paper and pencil environment, teacher's work for preparation may be greater than before.

From T1 to T4, the role and place offered to the teacher in the environments increase. Hopefully, it does not necessarily means that, from T1 to T4, the use of TEL environments is heavier for teachers. In a certain extent, TEL environments from T4 could be used as microworld like in T1 without a huge implication of the teacher. The

questions arisen in paragraph about T4 about the importance of teacher's competencies are very important and could be reconsidered for all the categories about general teacher's technology competency and teacher's implication needed. The usability and the chance for TEL environments to be adopted and used are tied to the answers to theses questions.

If we consider taxonomies of teachers, from Canada [BIB-06] for example, there exists innovative pioneers, but also resistant to new technology, and in between those who are following the group with more or less scepticism and those who are only following the official directives when some are given by government. A mapping may be done from these categories onto our categories about TEL environments. Every teacher could have some TEL environment adapted to him. Hence, from that mapping one could imagine the number of teachers which would be interested by TEL environment from each category.

Finally, from the set of 21 TEL environments selected for a study in the research community [TEL-06], mostly in mathematics, and in the most advanced commercialised TEL environments in France, we have found out 5 environments belonging to T1, 4 environments belonging to T2, 5 environments belonging to T3 and 7 environments belonging to T4. This is not necessarily representative of all TEL environments designed and used. In fact, we think that our study privileged rich TEL environments. From our research point of view, these are the most interesting ones. From pedagogical point of view, the main interest could be for another group.

2.2 Activities foreseen and provided by TEL environments

The second short taxonomy is about the activities which are foreseen and provided by TEL environments. We point out technical and technological competencies needed for these activities and compare these activities done in TEL environments contexts with similar activities accomplished without computer in the regular work of the classes. Our goal is to find what could be the added value brought by computers.

From a global point of view three main activities are considered:

- **Lessons**. From our point of view (TEL environments designer and developer), the organisation, with a computer, of such learning activities is not the most difficult one. Moreover, in a large part, it is not specific to learning environment. Modern technologies and modern programming languages offer components which permit to easily display and edit multimedia documents with texts, mathematics, images, videos… Even organizing or assembling statically lessons for the construction of curricula is quite easy, or not so difficult. Difficulties and specificities appear when the organisation and the assembly are done automatically, like in [MEL-01]. But this is a service which is rarely proposed, and, which still relies heavily on the availability of the content.

  Our opinion is that the most difficult part related to that activity does not concern designers and developers of TEL environment but people – teachers, didacticians or mathematicians for our concern – who define and create the content of such documents. Moreover, most of the work related to lessons and the way lessons are performed is not specific to TEL environments, it is analogous to the one done without computer.

- **Discovery and Training activities**. From our point of view, discovery activities and especially training activities introduce important works for designers and developers of TEL environments. When this is well done, it consequently allows added-values in comparison to classroom without computer activities. For example: feedback, hints, randomly generated exercises…

  One of the major difficulties in our domain (math education) in the design of TEL environments for discovery and training activities come from the necessity for the TEL environment to tackle the complexity of the domain (here mathematics.) We mean, complexity with the representations used, complexity with the semantics of the manipulated objects, complexity with the mathematical rules applied on these objects. One solution consists to link the TEL environment with a math expert: a CAS (Computer Algebra System) and to rely on the CAS for all the mathematical questions. That solution has drawbacks: Will the expert be understandable and understood? How the expert will be controlled? Can we give full freedom to the expert? It is not clear that this solution is the best one. We subscribe to the controversies about the ability of CAS to be the mathematical expert of a TEL environment [PRA-01]. Another solution is to reduce the training activities to multiple choice questions, or activities without any mathematical added-values, depending on the teacher for feedback, hints and creation of exercises.

  If we focus on the feedback provided during training activities when students have to practice with exercises to acquire techniques, automatisms and autonomy, we can consider the example of our software Aplusix. We have already pointed out for Aplusix the two fundamental feedbacks, the first one is about semantics, it concerns the equivalence checking, the second one is at crossed-road of syntax and semantics, and it concerns the validation of the end of an exercise. About syntax and semantics considerations, see [NIC-01]. The importance of syntactic and semantic feedbacks is essential. Here, the difficulty is that pedagogical and didactical elements may interfere with mathematical elements. Should the software accept mathematically correct answer when these answers are not pedagogically correct? And, how to express the didactical and pedagogical elements to be verified?

- **Test activities**. Test activities necessitate also important works for designers and developers of TEL environments but consequently allow added-values in comparison to classroom without computer activities, here, for example: automatic mark and diagnosis…

  About automatic marking, most of the difficulties share the same nature that those seen previously about the design of feedback in training and discovery activities: it is about the necessity for the TEL environment to tackle the complexity of the domain (for us mathematics) and it raises questions about the use of an expert (for us a CAS environment) to achieve that goal.

  About diagnosis, difficulties and challenge do not concern mathematics but learning theories, didactics and pedagogy and cannot be solved by general didactical or pedagogical software solutions, because there is not any one available. The domain seems to be much more complex than all those seen previously. And it is not clear that a general diagnostic algorithm of human learning will be found even if some general theories tend toward that goal [SIS-98]. Only few local solutions are available, and these local solutions have necessitated a lot of work, see for example systems based on ACT theory [AND-04].

On top of these activities, we must also notice the need for functionalities in TEL environments for managing students, classes, lessons, exercises and tests, and for sequencing the execution of such activities. Fortunately, there exists some general solutions for defining and playing scenarios composed of such activities, with selected students.

Considering usages, if we take into account what is working really in classes, from [BAL-05], out of the tree main activities seen above, the most important one, far ahead from the other is the discovery and training activity. In particular, activities where students have to work autonomously are pointed out in [BAL-05] as what can make the difference for a successful integration of new technology at school.

3 Building patterns of exercises

The last part of this communication is at the cross-road of the two previous taxonomies. It considers TEL environments in mathematics which provide teachers with tools for building exercises (or patterns of exercises) in the TEL environment in order to produce personalized training activities (or to generate randomly and dynamically lists of exercises). Three solutions can be observed:

- **Pedagogical Web Standard approach.** In other words, we are thinking about standards like IMS-QTI [IMS-04]. The advantage targeted by the use of such solutions is to gain compatibility and possible interoperability between different platforms and materials from different authors.

  The first main problem is that pedagogical standard like IMS-QTI does not know a lot about mathematics. To some extent, this is not so important when the only training activities concern multiple choice questions. But for open exercises, or unsolved exercises this could appear to be a major inconvenient. Some tentative exists to introduce Mathematics in standard [MathQTI], but generalisation to other domains are not available, and even the MathQTI is not sure to become a success used by every software.

  The second main problem is that this approach necessitates that the teacher willing to introduce new exercises, or pattern of exercises, must be familiar with the languages used like IMS-QTI, IMS-SS, IMS-LD, IMS-CP, LOM… Even for people aware of every new technology, this is a challenge. When user-friendly editors will be available, this solution could be practical, but it is not sure that in the next future user-friendly editors will appears. The languages IMS-* continue to evolve, for example in France for the LOM standard, evolution tends to add some specifics pedagogical elements or try to define the vocabulary used by index. One reason to understand why this approach is not 'naturally' suited for math teacher is to be found when we consider the goal of the IMS consortium: seems to be more oriented toward industries and institution than toward teachers.

- **Specifically Math web approach with CAS.** The most commonly used approach consists to integrate Computer Algebra Systems to open-source or proprietary web system, for ex. [Web+CAS]. In such systems, math teachers can write small algorithms to generate the mathematical part of the exercises. Of course it does necessitate knowing, using and mastering a programming language, usually the programming language of the CAS.

About the organization of exercises and pedagogical issues, web programming techniques are used (php, mysql…). Fortunately, these parts of the system can be and usually are hidden to math teachers, and the system behaves with default process.
Last, but not least, about presentation and representations of the content and the math content, when editors are not available, math teachers have to know a little about web language (html, css…) but are then free to present their activities attractively, with animation, colours...

The major inconvenient of that approach is that it does necessitate that the teacher willing to introduce new exercises or pattern of exercises must be familiar with technologies and programming languages used by CAS and Web platforms. Even if CAS technologies and programming languages are more familiar to math teachers and technologies and programming languages of web platform are much more widespread in our society that approach still concern specialists up to date with new technologies.

- **(Our) Teacher-friendly approach.** Our approach tries to avoid the burden of technology by giving teacher, first, easy to use and efficient tools to design exercises and pattern of exercises, and second, by trying to help the teacher with intelligent functionalities which would give him feedback about what s/he has done, and what will be the result.

  The first element of that approach is the design and realization of an editor for pattern of exercises and map of exercises. For exercises, the Aplusix environment already contains an exercise editor. It will not be described here. For pattern of exercises, our work relies on the work already done in Aplusix to give a definition and a structure to the global object 'pattern of exercises' [BOU-05b]. It leads to the previously described map of exercises with patterns of exercises. Until now, these patterns were hidden and unreachable. Teachers were asking for some control on it (visualize them, edit them), so we have worked on the design of a user friendly interface for teachers.

  Two ideas have conducted our work. The first idea was to try to escape the use of formal language or programming language: whether responses are given in some natural language or responses are to be selected from predefined lists of items. The second idea is to provide the teacher with examples of instantiation of the pattern as soon as enough elements are defined (see right panel, figure 4). This is the major part of the second element of our approach. In fact, we try to give immediate non intrusive feedbacks. This follows the same ideas as in the main part of the Aplusix environment, the part devoted to students. Another feedback is also available: a rough evaluation of the number of different instantiated exercises produced with the current pattern. We also thought about an automatic calculation of the solution of the instantiated exercises, and of the complexity of the resolution (with the number of elementary steps to be performed to achieve a solution) but we did not achieve that part.

  We also added the possibility to use Aplusix with the new pattern of exercises as soon as that pattern is 'correct', in the same condition as a student would do. And last, the teacher can organize his patterns in a tree-like manner, and reutilize patterns available in Aplusix.

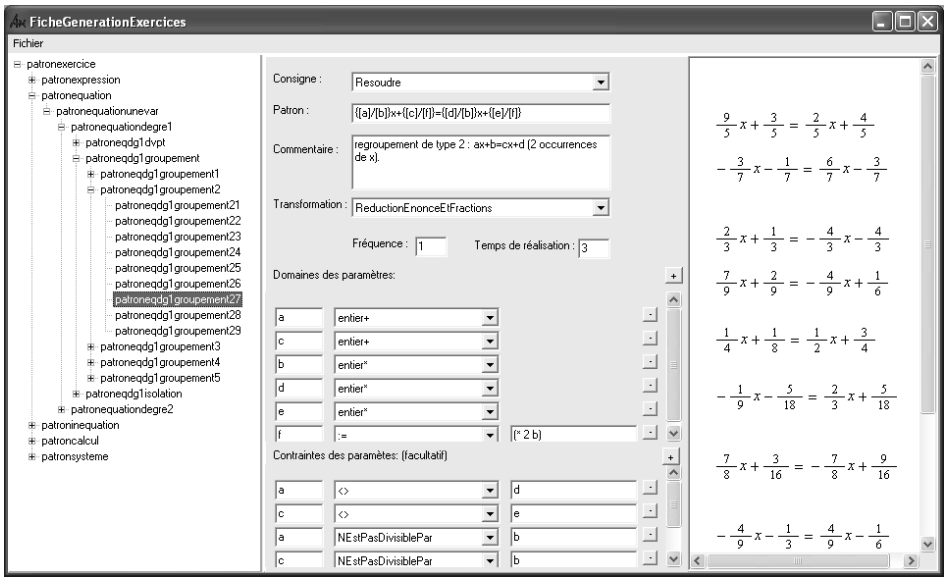

Figure 4 Editing patterns of exercises in Aplusix (prototype).

4 Conclusion

On the occasion of the design of a prototype for a pattern editor in Aplusix, we have studied: first the role and place given to teachers in TEL environments, second, the activities foreseen in TEL environments, and at the cross-road, ways TEL environments propose to build exercises and patterns of exercises. From these studies, we have observed the influence of learning/teaching theories (behaviourism, constructivism, socio-constructivism…) According to us, such influence is due partly from the design of TEL environments and there are ways to liberated usages from it when TEL environments are been planned to cope with a lot of situations, and even permit free activities.

About the pattern editor, patterns of exercises were not new in Aplusix and generated exercises were appreciated by teachers, but patterns were unreachable and teachers asked for some control on them, so we have worked on a prototype of an editor. This is a first step toward an editor of pattern of exercises which could be integrated in the next version of Aplusix.

About generating exercises with random coefficients, two perspectives for the future can be imagined. The first one would be to try to extract the didactical variables used to define the hierarchy of patterns of exercises existing in the map of exercises of Aplusix in order to build a new generator of exercises whose input would not be patterns itself but didactical variables. The second one would be to find a way to introduce didactics in the pedagogical construction imagined in the IMS-like standards.

About usage, the new prototype permits the definition of two medium sized (fifteen families of exercises, more than fifty patterns), but fully usable, maps of exercises of arithmetic at elementary schools in few hours (less than 3h.) This is to be compared to the

number of days, and even weeks, the first map (thirty eight families of exercises, more than three hundreds of patterns) necessitated.

The building of the editor introduces a shift in the way patterns of exercises and map of exercises are organized and taken into account in Aplusix, but this is another story.

References

[AND-04] Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., and Qin, Y . "An integrated theory of the mind", in *Psychological Review* Vol. 111, No 4, 2004.

[BAL-05] « L'elearning dans les pays baltiques » in *Rapport de mission de veille et d'étude 8-15 mai 2004*, Direction de la technologie/SDTICE/réseau international, Centre international d'Etude Pédagogique, 2005.

[BIB-06] Bibeau, R. «La vie avec les TIC, la vie après les TIC». In *Revue de l'association EPI.* Paris. Octobre 2006.

[BOU-05a] Bouhineau, D., Nicaud, J.F., Chaachoua, H., Bittar, M. and Bronner, A.: "Two Years Of Use Of The Aplusix System" *in proceedings of the 8th IFIP World Conference on Computer in Education*. Le Cap, South Africa, July, 2005.

[BOU-05b] Bouhineau, D., Bronner, A., Chaachoua, H., Mezerette, S. and Nicaud, J.F.: "Computer Assisted Assessment in Elementary Algebra, Experiences and points of view from the APLUSIX project", in *Maths CAA Series, LTSN Maths, Stats & OR Network*, University of Birmingham, UK, November 2005. mathstore.ac.uk/articles/maths-caa-series/nov2005/

[CHA-04] Chaachoua H., Nicaud, J.F., Bronner, A. and Bouhineau, D.: "APLUSIX, A learning environment for algebra, actual use and benefits" i*n proceedings of ICME'10 : 10th International Congress on Mathematical Education Copenhagen*, Denmark. July 4-11, 2004.

[IMS-04] IMS Question and Test Interoperability: "Item Implementation Guide" & "Item Information Model", V2.0 Editor. S. Lay, Univ. of Cambridge, http://www.imsglobal.org/question/qti_item_v2p0pd written in June 2004.

[MathQTI] http://www.maths.ed.ac.uk/mathqti/

[MEL-01] Melis, E., Büdenbender, J., Andrès, E., Frischauf, A., Goguadze, G., Libbrecht, P., Pollet, M., and Ullrich, C.: "ActiveMath: A Generic and Adaptive Web-Based Learning Environment" in *International Journal of Artificial Intelligence in Education*, Vol 12, Nª4, 2001.

[NIC-01] Nicaud, J.F., Bouhineau, D., and Gélis J.-M., "Syntax and Semantics in Algebra", in *Proceedings of the 12th conference of the International Commission on Mathematical Instruction (ICMI) : The Future of the Teaching and Learning of Algebra*, Eds Helen Chick, Kaye Stancey, Jill Vincent & John Vincent, University of Melbourne, Australia, 2001.

[NIC-04] Nicaud, J.F., Bouhineau, D., and Chaachoua H.: "Mixing microworld and CAS features in building computer systems that help students learn algebra", in *Int^{al} Journal of Computers for Mathematical Learning*, Vol. 9, Issue 2, Springer-Verlag, 2004.

[NIC-06] Nicaud, J.F., Bouhineau, D., Chaachoua H. and Trgalova J.: "Developing Interactive Learning Environments that can be used by all the classes having access to computers. The case of Aplusix for algebra" *in the Seventeenth ICMI Study: Technology Revisited*, Hanoi, Dec 2006.

[PRA-01] Prank, R. and Tõnisson, E.: "Is the Domain Expert Module for Expression Manipulation Exercises Ready?" *Proceedings of the Tenth International PEG Conference*, Tampere, 2001.

[SIS_98] Sison, R. and Shimura, M., "Student Modeling and Machine Learning" *in International Journal of Artificial Intelligence in Education*, Vol. 9, 1998.

[TEL-06] "Analysis of a selection of educational ICT systems for mathematics", deliverable 20.5, telma.noe-kaleidoscope.org, 2006.

[Web+Cas] http://wims.unice.fr/wims/ http://maths.york.ac.uk/aiminfo/