Automatic Multi-criteria Assessment of Open-Ended Questions: A Case Study in School Algebra

Élisabeth Delozanne[1], Dominique Prévit[2], Brigitte Grugeon[3], and Françoise Chenevotot[3]

[1]L'UTES - Université Paris VI - 4 pl. Jussieu - 75005 PARIS – France,
elisabeth.delozanne@upmc.fr
[2]LIUM-Avenue Laennec 72085 Le Mans Cedex 9– France,
dominique.previt@bretagne.iufm.fr
[3]DIDIREM - Université Paris VII - 2 place Jussieu - 75251 PARIS Cedex 5 – France,
brigitte.grugeon@amiens.iufm.fr,
francoise.chenevotot@lille.iufm.fr

Abstract. This paper deals with authoring assessments of complex competence involving open-ended questions. We present, PépiGen, a multi-criteria automatic assessor for school algebra, via a walkthrough of an example. PépiGen is based on our previous work on Pépite, an automatic cognitive diagnosis tool that capitalizes on educational research results. From that prototype, we derived patterns of diagnosis tasks. A pattern models (i) a class of exercises, (ii) the different students' points of view on the solutions reported in the literature or observed in a corpus, (iii) and a multidimensional assessment for each solution approach. To adapt an assessment to a specific classroom context (e.g. level of difficulty, time, learning objectives) an interface allows an IT non expert (e.g. a teacher) to generate new instances of exercises by filling the pattern parameters. The originality of our research lies in the fact that our system generates the automatic analysis of students' simple or complex answers, such as algebraic reasoning. This is an ongoing work but preliminary evaluation shows that PépiGen is already successful in generating and analyzing most answers on several classes of problems.

1 Introduction

The work reported here is part of an ongoing project, the Lingot project. Its objective is to design an intelligent aid that supports math teachers when they have to monitor learning in a classroom context, taking into account their students' cognitive diversity. This paper focuses on diagnosing students' cognitive profiles in algebra. It presents PépiGen, a system that generates Automatic Multi-criteria Assessments of students' competence in school algebra.

We first present the background, the objectives and the methodology we adopted to elicit patterns from the first Pépite assessment system used as a prototype. Then, we illustrate the modelling language we defined by describing an example of pattern of diagnosis tasks involving open-ended questions. The next section describes PépiGen, the system that allows a user to generate diagnostic tasks that instantiate patterns. We

B. Woolf et al. (Eds.): ITS 2008, LNCS 5091, pp. 101–110, 2008.

end with a discussion of our work in comparison with related works and with a summary of contribution and plans for future research.

2 Background

The key point of our assessment approach is that students' answers to problems are not simply interpreted as errors or as lack of skills but as indicators of incomplete, naive and often inaccurate conceptions that the students themselves have built. A fine analysis of the students' work is required to understand the coherence of the personal conceptions, to develop or to strengthen right conceptions, and to question wrong or unsuitable ones that interfere with, and sometimes prevent learning [1]. Detecting these conceptions is a very complex task that requires special training and a lot of time. ITSs can be a very helpful aid for teachers to reveal implicit conceptions which are very difficult to access without automatic reasoning on students' performance. Designing such systems is not trivial; especially when the student's input is not very constrained.

We developed such a cognitive diagnosing tool, derived from Educational Research [6], called Pépite, and we tested it in real settings [3]. This previous work aimed to prove that it was possible to automatically build a rich student cognitive profile from data collected after the student solved a set of tasks especially designed for that purpose. These tasks involved preformatted answers and open-ended answers. Like in other systems [5], in Pépite, the diagnosis is a three stage process. First, a local diagnosis provides, for each student's answer, a set of codes referring to the different criteria involved in the question. A code gives an interpretation of the student's answer according to a set of 36 *criteria* on six *assessment dimensions* (see section 4 for an example). Second, Pépite builds a detailed report of the student's answers by collecting the same criteria across the different exercises to have a higher-level view on the student's activity. At this stage, the diagnosis is expressed by success rates on three *components of the algebraic competence* (usage of algebra, translation from one representation to another, algebraic calculation) and by the student's *strong points* and *weak points* on these three dimensions. This level is called *personal features* of the student's cognitive profile. Third, Pépite evaluates a level of competence in each component with the objective to group of students with "equivalent" cognitive profiles. This level is called the *stereotype* part of students' profiles. Stereotypes were introduced to support the personalization in the context of whole class management and to facilitate the creation of working groups [4].

3 The PépiGen Project

In the present stage of the project, the aim is to offer an authoring tool, called PépiGen, to generate different Pépite-like diagnosis tools adapted to different school contexts and teachers' objectives. We had a lot of feedback from teachers who used the previous Pépite tools [3]. One of their points was that Pépite was interesting for a given school level. But teachers would need a database of diagnosis exercises to use Pépite-like tools at other school levels. Most teachers asked for off-the-shelf diagnosis

material, arguing that their job was to monitor learning, not to author materials. Some asked for assessments that can be tuned to specific contexts. Very few asked to define their own exercises but they asked to do so with no programming at all. These observations are confirmed by [16] in a state of art review of ITS authoring tools.

Thus, the work reported here describes how to build *banks of exercises supporting the diagnosis*. We focussed on the following design scenario: a teacher chooses a prototypic exercise in the bank and, if need be, asks for another equivalent one retrieved from the bank, or adapts the statement of the exercises by filling in forms (Cf. 6.1). In order to achieve this objective, in this paper, we investigate two research questions:

1. How to derive patterns of diagnostic tasks from the first Pépite prototype?
2. How to generate the procedure to analyze open-ended questions when (most) current technology restricts to preformatted answers?

From a computational point of view, the most difficult problem to be solved was to design and implement a system that assesses open-ended answers, both generic enough to apply to many classes of algebraic problems, and specific enough to detect students' personal conceptions. With open-ended questions, it is impossible to predict every student's answers. Thus the main points in our design are (i) to anticipate most current students' solution approach to one type of question by detailed and accurate epistemological and empirical studies, and (ii) to generate a set of answers representing each solution approach.

Our research approach is a bottom-up approach informed by educational theory and field studies. In previous work, we started from a paper and pencil diagnosis tool grounded in mathematical educational research and empirical studies [1, 6]. Then we automated it in a prototype called Pépite and tested it with dozens of teachers and hundreds of students in different school settings [3]. In the present research, we generalize this first design to create a framework for authoring similar diagnosis tools offering configurable parameters and options.

4 An Example of Diagnosis Task Pattern

Let us take a prototypic exercise from the original Pépite involving an open-ended question (Fig. 1). The objective of this exercise is to have deep insight in the student's algebraic thinking and to assess her/his skills and conceptions in the six dimensions of algebraic competence: (i) Validity, (ii) Meaning of Letters, (iii) Algebraic Writing, (iv) Translation (ability to switch between various representations: graphical, geometrical, algebraic, natural language), (v) Type of Justifications ("proof" by example, proof by algebra, proof by explanation, "proof" by incorrect rule), (vi) Numerical Writing.

Table 1 shows four examples of students' answers and their coding in Pépite. In those examples we can notice that no students' solutions are fully correct, but we can suspect very different levels of development in their algebraic thinking. Of course, building a cognitive profile from one answer is not reliable, but we can hypothesize that these students will benefit from different learning activities [4].

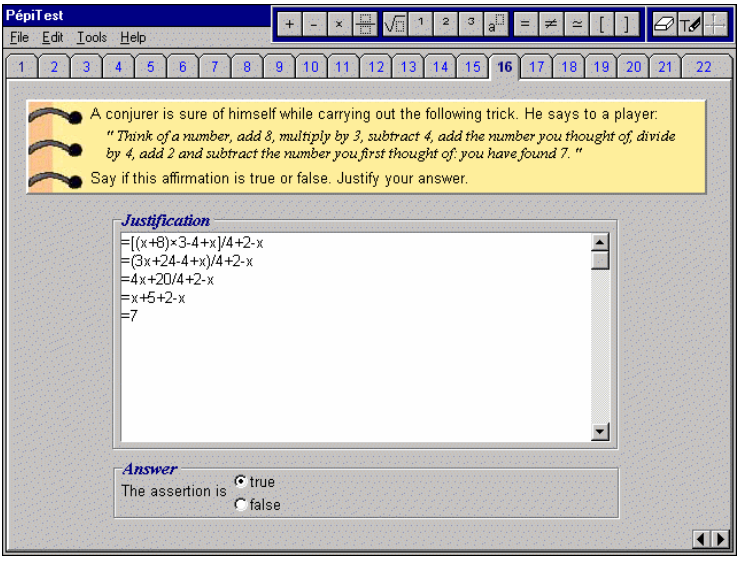


Fig. 1. A Pépite prototypic exercise of the "Proof and calculation process" pattern

To clone this exercise for a lower school level, we considered the following design scenario. An author is presented with the prototypical exercise and changes the italic sentence in the statement by the following one (statement 2): *Think of a number. Add 6 to this number multiply the result by 3, subtract three times your number to the result. You find 18.* This statement is a parameter of the pattern.

The system generates the algebraic expression, here (x+6)*3-3*x. The difficulty is to generate the anticipated solutions and their coding. In this type of task, [6] distinguished mainly four approaches for students to justify their answer:

1. An algebraic approach involving several processing types
   a. A correct translation in algebra by a global expression with correct/ incorrect use of parenthesis and an optimal-correct/non optimal correct/incorrect reduction to a number (7 or 18 in the examples);
   b. A partially correct translation to algebra using a step-by-step translation with correct/incorrect reduction to a number;
   c. An incorrect translation where the equal sign is not an equivalence sign between numbers.
2. A numerical approach where the student takes one or several examples involving the same types of processing as in the algebraic one;
3. A combination of both approaches where the student tries an algebraic proof but does not succeed and falls back on numerical examples to justify;
4. A justification in natural language.

In Table 1, Laurent's and Karine's solutions are examples of the first approach, while Khemarac's and Nicolas's are examples of the second one. For each solution approach and processing type, PépiGen, generates a corresponding set of algebraic

Table 1. Types of students' answers and their multidimensional coding in Pépite (age 15 or 16)

Khemarak	Nicolas	Karine	Laurent
Soit 5 un nombre ((5+8)×3-4+5)/4+2-5=7 ? ((13)×3-4+5)/4+2-5=7 ? (39-4+5)/4+2-5=7? 10+2-5=7 ? 10-3=7 ? 7=7 ? Oui donc cela marche (Yes thus it works)	3 + 8 = 11 11 × 3 = 33 33 - 4 = 29 29 + 3 = 32 32/4 = 8 8 + 2 = 10 10 - 3 = 7	x + 8 = 8x 8x 3 × 8x = 24+3x= 27x 27x-4 = 23x 23x+x=24x 24x/4=6x 6x+2=8x 8x-x=7	=[(x+8)×3-4+x]/4+2-x =(3x+24-4+x)/4+2-x =4x+20/4 + 2-x =x+5 +2-x =7
Justification by example (J2)	Justification by example (J2)	Justification by school authority (J4)	Justification by algebra (J1)
Valid translation in algebra (T1). Global expression with parenthesis, expressions are seen as a whole	Partially valid translation(T2). Step- by-step translation, expressions are seen as a process	Algebra is use to abbreviate (T4). The = sign announces a result, not an equivalence	Valid translation in algebra (T1). Global expression with parenthesis, expressions are seen as a whole
Correct numeric writing rules (NWR1)	Correct numeric writing rules (NWR1)	Incorrect identification of operation (AWR4); incorrect algebraic rules : $x + a \rightarrow x\ a$ $a\ x \pm b \rightarrow (a \pm b)$ $a\ x - x \rightarrow a - 1$	Incorrect use of parenthesis with memory of the meaning (AWR31)
No use of letters (L5)	No use of letters (L5)	Use of letters to calculate with incorrect rules (L3)	Correct use of letters (L1)
Invalid answer (V3)	Invalid answer(V3)	Invalid answer (V3)	Invalid answer (V3)

expressions. It associates a set of codes that characterizes the algebraic processing type from a diagnosis point of view.

One pattern describes the original exercise and the exercise generated by statement 2. The pattern *name* is: "Proof and calculation process". The two exercises are "similar" because the *interface*, the set of words to express the statement (see the "palette" Fig. 2), the diagnosis *objective*, the *anticipated solving approaches*, and the *set of possible codes* involved are all the same.

The differences between a clone and the prototypic exercise are the statement, the algebraic expression that translates the statement in algebra, and the complexity of this algebraic expression (level of parenthesis, number of operators, and number of division). The statement and the algebraic expression are *parameters* of the patterns and the three indicators for the complexity are *parameter characteristics*. These characteristics will be used to query the database and to tune a test to a school level. The parameters may be constrained. In the example, there is one *constraint*: the algebraic expression is reduced in a constant or a linear function; otherwise the diagnosis task

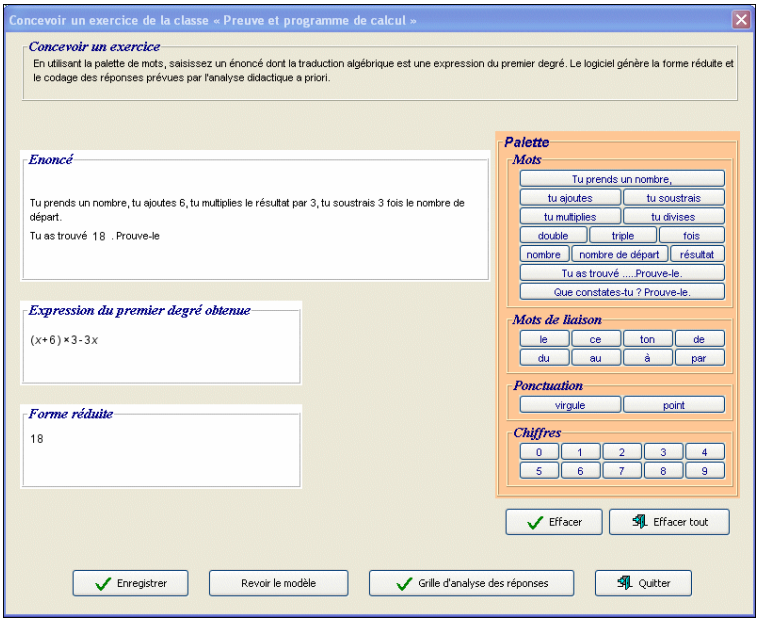


Fig. 2. Parameters setting for the "Proof and calculation process" patterns of diagnostic task

would change. The differences in the diagnosis part of the exercises are expressions representing *optimal correct solutions*, *non optimal correct solutions*, *partially correct solutions*, *incorrect solutions*. Each solution is characterized by a *comment*, a *code*, one or several *expressions* and correct or incorrect *rules*.

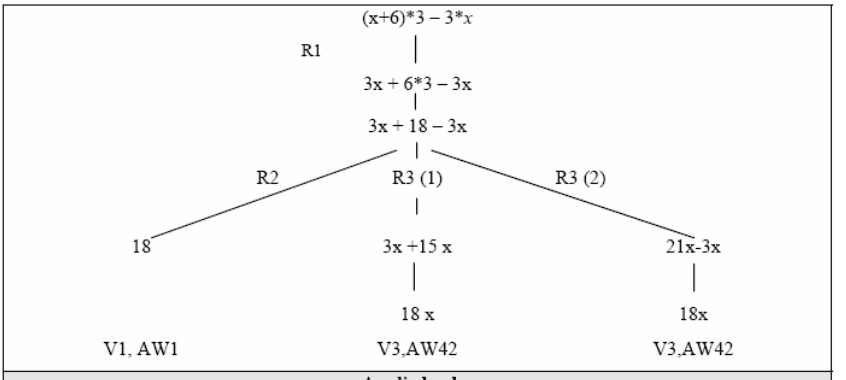
5 How to Generate a Diagnostic Task from a Pattern?

PépiGen is implemented in Java. It creates, initializes and saves, in an XML database, instances of the different classes representing the dynamic part of a pattern of diagnostic tasks. The static part is described by an XML schema. A *diagnostic task* consists of an exercise (problem statement and questions), a set of correct or incorrect anticipated solutions, and a set of codes that characterizes each solution from a cognitive diagnosis point of view. It is generated by PépiGen once the parameters of a pattern are set. Thus generating a diagnostic task is a two stage process: setting the parameters and generating the solutions tree and the coding for each branch. Data generated are stored in XML files and retrieved at run time to generate the student interface and to assess the student's answer.

When very constrained, the parameters are automatically generated by PépiGen (e.g. a formula to be instantiated with integer values between 1 and 20). This mode is called *automatic parameter setting*. But, for more complex patterns, the parameters are set by a human author (a teacher, a teacher trainer or a researcher). This mode is called *aided parameter setting* (e.g. Fig. 2).

When human authoring is required to set the parameters, PépiGen provides a Graphical Interface to enter the parameters. The author enters one parameter, the statement in natural language using the palette on the right side of the screen, and PépiGen generates the other parameters (the corresponding global algebraic expression and its reduced form), and displays them on the left of the screen. A software component based on a grammar and a finite state machine is used to interpret users' input in a constrained natural language and to translate it into algebra. This component is also used for analysis of students' input in other diagnosis tasks.

When parameters are set, a procedure specific to the pattern is called by PépiGen, to automatically generate all the information necessary to diagnose the students' answers to the exercise. This procedure is simple when answers are preformatted. In case of open-ended questions involving the dimensions "Algebraic calculation" or "Numerical calculation" in the pattern description, a software component, called Pépinière, builds a tree representing all anticipated solutions to the exercise and codes each solution on several dimensions.



Applied rules		
Correct rules	R1 : (A+B)C → AC+BC R2 : AC+BC→ (A+B)C	
Incorrect rules	R3 : A+BC→ (A+B)*C Applied to : (1) 18 – 3x (2) 3x +18	
Codes		
Validity	V1 : correct	V3 : incorrect
Algebraic Writing	AW1 : Correct use of rewriting rules	
	AW42 : The student gathers the terms in an algebraic sum.	

Fig. 3. Anticipated algebraic solutions for the clone example

Pépinière is a specific Computer Algebra System (CAS) dedicated to interpreting and generating students' algebraic input according to an epistemological and didactical analysis. It is independent of the different patterns. It relies only on mathematical foundations (mainly parsing of mathematical expressions, unification theory, algebraic rewriting rules), and on the multidimensional model of algebraic competence that grounded the Pépite project (i.e. on a set of multidimensional criteria represented

by a code, an extensible structured set of rewriting rules and a set of heuristics to prevent infinite loops). [14] presents a detailed description of Pépinière. In the present section, we just describe our general approach to automatic generation of the diagnosis by illustrating it with one example of pattern instantiation. In this example, the procedure to generate the coding instructions file is a two step process.

First, Pépinière builds a tree with every anticipated solution. It applies correct and incorrect reduction and developing rules. Heuristics are used to tackle the difficult problems of combinatorial explosion and infinite loops [14]. Fig. 3 shows the tree generated from the algebraic expression parameter that characterizes the clone (x+6)*3-3*x.

Second, the tree is walked in a way specified by the type of approach (algebraic/numeric). Each node (expression and rule applied) is saved along with the coding. For instance, correct solutions are generated by saving the nodes in walking through the tree considering only the correct rules. Incorrect solutions with an algebraic approach and a correct translation to algebra are generated by saving the nodes with incorrect rules. For incorrect solutions with a step-by-step translation Pépinière is called recursively with expressions generated by the preceding step. Incorrect solutions with a numerical approach are generated in the same way.

After students passed the test, the diagnosis system asks Pépinière to compare one expression in the student's answer to the expressions in the coding prescription file. To this end, Pépinière builds trees representing the expressions and tests the equivalence of the expressions regarding the commutability and associability of the operators.

6 Tests

Since PépiGen is still in the development phase it is difficult to have usability tests in real settings with teachers. Thus, we describe here a primary evaluation round. First we tested PépiDiag on a corpus of answers collected with the prototypic exercise (N=353) and its clone (N=39) presented in section 5. The system coding was validated by two educational researchers (the third and fourth authors). They agreed 100%. This means that PépiGen implementation is conform to the educational research model PépiGen is based on. Then, we asked three mathematics teachers to generate clones with PépiGen. They understood the potential of the system and found it easy to create exercises. They were satisfied with the solutions generated. We also tested Pépinière to generate solutions for other patterns involving simpler algebraic reasoning [17].

7 Related Work

Assessment and student modeling is a hot research topic in ITS and the e-learning community. We are especially interested in assessment modeling approaches and particularly in assessment of mathematical skills involving open-ended questions.

The leading specification for assessment is QTI, developed by IMS Global Learning Consortium [8]. The primary goal of this specification is interoperability between Learning Management Systems but it is limited to multiple-choice items and their

variations. [9] provide a broader conceptual model for assessment allowing the use of several assessment instruments (e.g. portfolio assessment or peer-assessment) and several types of assessment (e.g. multi-dimensional assessment). It is a first step to integrate QTI and IMS-LD specification. A perspective of our work could be to test their model by translating to their Item Construction Model, our conceptual model of diagnostic task patterns exemplified in section 5. But, so far it is unclear for us, if their model can represent both correct and incorrect conceptions. Moreover, as far as we know, it is a descriptive model and there is no implementation. In section 6, we presented through a worked example, a domain specific implementation corresponding to the "response rating part" of their model.

Many ITS or e-learning systems focus on math education and implement student's modeling or assessment authoring tools. Some of them analyse open answers when they are numerical or reduced to a single algebraic expression (Algebra Tutor [10], Assistment [2], LeActiveMath [11]). Very few analyse a whole reasoning. From this point of view, closely related to our work are Diane [7], Andes [15], and Aplusix [12].

Diane is a diagnosis system to detect adequate or inadequate problem solving strategies for some arithmetic classes of problem at elementary school level. Like Pépite, it is based on a very precise cognitive analysis. For each isomorphic class of problems, Diane analyses open-ended numerical calculation according to several criteria. It is very efficient compared to human assessment by experts. However, for more complex domains such as Physics or Algebra, researchers had to use a standard CAS or to develop one, specific to the type of students' inputs and to the type of diagnosis needed in the project.

For instance, Aplusix is a micro-world devoted to algebra learning in secondary schools, widely used in actual classrooms in France and in other countries. A teacher generates problems from different patterns of algebraic expressions for several tasks (e.g. factorisation, equation). Aplusix provides a very fined grained analysis of students' use of algebraic rewriting rules. PépiGen diagnosis is not so deep in the algebraic writing dimension but assesses a broader panel of skills on five other dimensions because the objective is to link formal processing with other students' conceptions like meaning of letters or meaning of algebra. Thus, in the Lingot project, there are very different diagnosis tasks involving algebraic expressions but also geometric figures and calculation programs.

8 Conclusion

In this paper we presented an approach to design and implement Automatic Multi-criteria Assessment of open-ended questions in early algebra. Our approach balances between very specific and rigid off-the-shelf tools and heavy generic authoring tools [16]. We benefited from empirical and theoretical educational studies to model patterns of diagnostic tasks. We designed and partially implemented the PépiGen system that automatically generates the diagnosis tasks after the parameters have been set. A specific CAS, Pépinière, generates all the students' reasoning usually observed in math class and assesses them with multi-dimensional criteria. PépiGen is a significant step toward an interactive assessment authoring tool in Algebra to support teachers in addressing their students' difficulties more effectively. Although the first PépiGen

testings are encouraging, there is still much work to be done. We are currently completing the system development by implementing automatic diagnosis on reasoning on other classes of algebraic problems (e.g. equation solving). We are also investigating with educational researchers how learners themselves can benefit from the Pépite diagnosis.

The software component we implemented to analyze answers to open-ended questions is inevitably domain dependant, but we propose a model to describe pattern of diagnosis tasks derived from educational research that could apply to many problem solving assessments using explicit criteria on several dimensions of evaluation.

References

1. Artigue, M., Assude, T., Grugeon, B., Lenfant, A.: Teaching and Learning Algebra: approaching complexity through complementary perspectives. In: ICMI Study Conference, Melbourne, pp. 21–32 (2001)
2. http://www.assistment.org/
3. Delozanne, É., Prévit, D., Grugeon, B., Jacoboni, P.: Supporting teachers when diagnosing their students in algebra. In: AIED supplementary proceedings, pp. 461–470 (2003)
4. Delozanne, É., Vincent, C., Grugeon, B., Gélis, J.-M., Rogalski, J., Coulange, L.: From errors to stereotypes: Different levels of cognitive models in school algebra. E-learn, 262–269 (2005)
5. Delozanne, É., Le Calvez, F., Merceron, A., Labat, J.-M.: A Structured set of Design Patterns for Learners' Assessment. JILR 18(2), 309–333 (2007)
6. Grugeon, B.: Design and development of a multidimensional grid of analysis in algebra. RDM 17(2), 167–210 (1997)
7. Hakem, K., Sander, E., Labat, J.-M.: DIANE, a diagnosis system for arithmetical problem solving. AIED, 258–265 (2005)
8. IMS Question & Test Interoperability, http://www.imsglobal.org/question/index.cfm
9. Joosten-ten Brinke, D., van Bruggen, J., Hermans, H., Burgers, J., Giesbers, B., Koper, R., Latour, I.: Modeling assessment for re-use of traditional and new types of assessment. Computers in Human Behavior 23(6), 2721–2741 (2007)
10. Koedinger, K.R., Anderson, J.R.: Intelligent Tutoring Goes to School in the Big City. IJAIED (8), 30–43 (1997)
11. LeActiveMath project, `http://www.leactivemath.org/`
12. Nicaud, J.F., Chaachoua, H., Bittar, M.: Automatic calculation of students' conceptions in elementary algebra from Aplusix log files. In: Ikeda, M., Ashley, K.D., Chan, T.-W. (eds.) ITS 2006. LNCS, vol. 4053, pp. 433–442. Springer, Heidelberg (2006)
13. Murray, T.: An Overview of Intelligent Tutoring System Authoring Tools: Updated analysis of the state of the art. In: Authoring Tools for Advanced Technology Learning Environments, ch.17, Kluwer Academic (2003)
14. Prévit, D.: PépiGen, un générateur de batteries d'exercices pour un diagnostic cognitif en algèbre élémentaire, PhD dissertation (to appear)
15. Shapiro, J.: An Algebra Subsystem for Diagnosing Students' Input in a Physics Tutoring System. IJAIED 15, 205–228 (2005)