Learning with virtual agents: Competition and Cooperation in AMICO

Dorothée Rasseneur[1], Elisabeth Delozanne[1,2], Pierre Jacoboni[1], Brigitte Grugeon[3, 4]

[1]LIUM, Avenue Olivier Maessian, 72085 LE MANS Cedex 09, France
{dorothee.rasseneur, elisabeth.delozanne, pierre.jacoboni}@lium.univ-lemans.fr
[2]IUFM de Créteil, Rue Jean Macé, 94861 BONNEUIL Cedex
[3]DIDIREM - Paris VII, 2, Place Jussieu, 75 251 PARIS Cedex 05, France
[4]IUFM d'Amiens 49, boulevard de Châteaudun 80044 AMIENS CEDEX,
brigitte.grugeon@amiens.iufm.fr

Abstract. This paper deals with the design and implementation of AMICO[1] , a prototype to enhance mathematics learning by interactions between a student (or a dyad) and virtual agents (several co-learner companions and a mathematician). This work is part of a multidisciplinary project, the project LINGOT that aims to give math teachers tools to improve their teaching. We first present the foundations of the project: the pedagogical assumptions, the design approach and the different kinds of virtual agents we propose. Then we present our virtual companions who allow students and teachers to use different learning and teaching strategies. They can play several characters: more or less trouble maker, more or less correct in their answers, more or less algebra speakers. In AMICO we propose two scenarios of interactions between students and virtual agents: competition and cooperation. We sketch the software architecture we propose in order to implement a simple authoring tool. Teachers specify problems, figures, the lists of common answers for students and companions. Then AMICO generates automatically and dynamically the dialogues between students and companions according to the companions' characters and students' answers. First user testing are encouraging and give perspectives.

1 Introduction

Learning with a virtual companion has been a focus of interest for ITS researchers for about ten years. Chan quoted that to bring up a future Chinese emperor an other boy was introduced to "learn with the prince" in order to improve the prince's learning. Chan proposed an ITS that allows students to learn with a virtual companion in the domain of indefinite integration. Since then, other Learning Companion Systems (LCS) have been created in different domains, although math is the most frequently

[1] AMICO sounds in French like an abbreviation for friendly companion and means Mathematical Learning by Interacting with Companions (in French: Apprentissage des Mathématiques par Interaction avec des COmpagnons)

Proceedings of ITS'2002, Biarritz (France), 5-8 juin 2002. Cerri S., Gouardéres G., Paraguaçu F. (eds.), Springer-Verlag, p. 61-70

chosen domain. Some systems support only one companion whereas others include both a virtual companion and a virtual tutor.
The aim of the multidisciplinary project Lingot is to provide mathematics teachers with software they can use in everyday classrooms in order to facilitate the teaching and learning of algebra. We agree with [10] that, from an educational perspective, the companions offer “a way to support the diversity of representations and perspectives inherent in human learning and teaching”. Indeed in order to help student to make sense to algebraic expressions some teachers organize classrooms debates. Educational research in mathematics has shown that, on some types of problems and with a careful class management, this pedagogical strategy enhances comprehension of algebraic expressions [6]. We based our software on this work. Grugeon claims that according to level of rationality the students are engaged in (proof by numerical example, proof by argumentation, proof by school authority, proof by algebra) they use different types of justification (counter examples, correct rules, mal-rules, examples etc.). Those justifications are articulated in several representation modes: algebraic, graphical, numerical, "mathural" language. By “mathural”, we mean a language created by students that mixes mathematics language and natural language: the formulations produced by the students in this language are not completely correct from a mathematical point of view but demonstrate a first level comprehension of mathematical notions [2]. To illustrate those table 1 shows an example of different justifications the students usually give and how those justifications are analysed in the educational research of Grugeon. Within this context, we assume that switching from one representation mode to another helps students to have a better understanding of algebraic expressions and to enter in a higher level of rationality.

Table 1. When stating that “$a^2 * a^3 = a^6$” isn’t true for every a, examples of several students’ proofs and their analysis in the model of algebra competencies that AMICO is based on

Level of rationality proof by …	Type of justification	Representation mode	Examples of students articulations
numerical example	To try with one or several numbers	numerical	“If I try with a = 1, 12 * 13 = 1 * 1 = 1 and 16 = 1”
argumentation	To give explanations	mathural language	“3*2 = 6 for exponents and not 2+3”
			“I have added the exponents”
school authority	To give rule relying on authority	mathural language	“We must never multiply exponents”
algebra	To take a counter-example	numerical	for a = 2, $2^2 * 2^3 = 4 * 8 = 32$ and 26 = 64
			“If we calculate for 2: $2^2 * 2^3 = 2*2 \ * 2*2*2 = 2^5$ and not 2^6”
	To give correct rule	algebraic	“for every a, we have $a^n * a^p = a^{n+p}$ and not a^{n*p}”
		mathural language	“To multiply powers of a same number, we add exponents”
			“When a number with an exponent is multiplied by this same number with an exponent, we add the exponents”

Our educational assumption is that an ITS based on interactions with virtual companions can improve the learning in inducing a socio-cognitive conflict. More precisely, we assume that to respond to the virtual peer's reactions, students may need to adapt their type of justification, and use several modes of representation. Our pedagogical goal is to find pedagogical scenarios to help the student in switching from one mode to another according to the situation, i.e. according to the problem or according to the interaction with the companion. In that context, our first objective is to test this idea by implementing AMICO.

For AMICO to be useful and usable by teachers in their classrooms, we have adopted methods of participative design based on the implementation of successive prototypes that are tested by students and teachers. This methodological choice led us to set a second objective: to define a flexible software architecture allowing fast modifications, flexible settings of parameters and automatic generation of the companions' interaction from data entered by teachers.

From those objectives come several research questions:

- As far as the pedagogical design is concerned:
  - Which interaction scenarios will induce socio-cognitive conflict and the use of several types of justification when a student (or a group of students) interacts with a companion?
  - What roles and features should the companions have in order to support different learning and teaching strategies? Is it necessary to introduce other virtual actors?
  - What types of relationships between human students and their virtual companions should be offered?
  - What learning situations can teachers set up when using AMICO in classrooms?
- As far as the software design is concerned:
  - Is the framework we propose powerful enough to allow teachers to adapt AMICO to the learning situations they want to set up?
  - What software architecture will be more effective to implement these situations and to generate instances of situations?

To begin to answer some of these questions, we introduce first the process that we adopted to find different types of companions and to design different interaction situations. Then we describe the "identity cards" of the companions we have designed in this prototype of AMICO and the different modes of interaction among the pedagogical actors. Next, we discuss the two scenarios proposed to elicit competition or cooperation between students and companions. Finally, we present the results of the tests we did and the teachers' opinions of our software.

2 The Design Process

As in our team's previous works [3] we based our work on a cognitive, epistemological and didactic analysis of a teaching problem. As mentioned above, we started from Grugeon's work [6] that has modeled the algebraic competence of students from (French) secondary schools. This model allows us both to analyse the set of exercises and to describe students' algebraic activity. Then we defined the context of use: AMICO is designed to be used in classrooms during individualised help sessions or when students work in dyads. The pedagogical objective is to have

students confront several types of justifications and to use different representation modes. We selected from exercises used in schools, those that to trigger of a debate, especially those relying on proof. For each of them we collected about sixty students' answers for use in AMICO. Our pedagogical design decision was that the virtual companions should be students' peers and should express in a student's style, often in incomplete or erroneous ways. Therefore we did not try to correct the collected answers. We agree with many researchers that students need to construct their own explanations. In our opinion learning will be better if students can discuss and examine various types of justification, the correct answer being revealed later by a virtual agent, the mathematician.
Based on ITS research dealing with virtual companions, we defined four types of companions and two learning scenarios. A first mock-up was tested by team members. A second was tested by six students (between 14 and 16) in the lab. This early testing allowed us to validate our interaction design and improve interface usability. Then the prototype was implemented and a new learning scenario added. Finally, this prototype was tested by ten teachers and trainers for mathematics teachers.

3 Different Companions Characters

Educational agents and learning companions have been described in the literature. They have various objectives, and do not have necessarily educational or accurate domain expertise.

3.1 The Different Types of Educational Agents

Chan [4] described three types of learning activities with virtual companions. The first one is "computer as a co-learner". Here, the companion is at the same knowledge level as the learner. They solve the problem together, but they also may be in competition. The virtual teacher or tutor does not intervene in this case. The second form of learning activity is the "learning companion system". A virtual teacher is added to facilitate the human and virtual students' learning. Students and companions solve in parallel the problem set by the teacher and then compare and discuss their various solutions. The third form of learning activity is the "learning by teaching". In this scenario, companions have a weaker level of knowledge than students, and so students have to teach the companion by giving him examples or information.
For the educational researchers we work with, it is important to let the students make incomplete or erroneous articulations so that their misunderstanding can be confronted. If teacher's intervention occurs too early the student can be left with a false or partial conception. This is one of the reasons why, from our point of view, virtual companions should express themselves as students, and not in a formal way. Therefore we decided not to introduce a tutor or a teacher in AMICO.
However, the experienced mathematics teachers all agreed it is of primary importance to end classrooms debates with a moment when the correct answer is stated and identified as mathematical knowledge and not as an opinion ("institutionalisation

phase”). To play this part, we introduced a virtual agent with the role of mathematician.
Educational researchers and teachers who share this conception on algebra learning and teaching agree with this design choice. It seems, although it remained to be confirmed, that teachers who have some more directive conceptions about teaching would wish to add a teacher agent who would explain and give several formulations to complete and correct the students’ or companions’ expressions.

3.2 Learning Companions in AMICO

[8] described research showing that not every student is satisfied by the same companion. Some prefer a very knowledgeable and self-confident companion; others prefer a less competent companion. We therefore decided to design different companions so that learners could choose according to their preferred learning strategies or teachers could select particular teaching strategies. [1] used some companions to disrupt learners deliberately by giving incorrect answers and by simply contradicting the learner. Such companions oblige students to confront theirs opinions with others and to justify their own ideas. In AMICO, we adopt this idea of disrupting companion. According to [11], better learning occurs when learners feel the need to teach something to their companion. Machine learning techniques could be used to implement the companion's learning. In AMICO, we did not explore this approach but we designed companions who make mistakes. We conjecture that students would want to explain their answers when they would think the companion was wrong.
Finally, to encourage understanding of algebraic expressions, our collaborating educational researchers recommended that students be incited to produce several types of justifications and to express these in several representation modes, in particular: algebraic language, natural language and numeric language. [7] estimated that the main difficulty for students when symbolizing word problems is that algebraic language is like a foreign language for the students. In their system, Ms Lindquist, they designed an artificial tutor who helps the students in developing this translation skill. In our project, it is a companion who incites the students to work in several representation modes. We designed a type of companion that almost always answers correctly but can use one of a number of representations. Choice of representation may be specified by the teacher. In AMICO two companions are of this type: the first almost always uses the algebraic language to justify his or her answers and the second uses the "mathural” language.

3.3 The Characters of the Pedagogical Agents in AMICO

In AMICO, we have chosen a range of features that can be parameterized by the teachers to define the characters of agents. The *mathematician* gives the correct mathematical answer to a question and a justification in an expression mode than can be specified by the teacher. For the companions, the first parameter is the degree of disrupting: it is the percentage of agreement or disagreement between the student’s and companion’s answers during a session. The second parameter is the frequency of use of different mathematical modes of representation (algebraic language, natural language, numeric language). The third concerns the companion's level of mathematical expertise which is represented by the percentage of correct answers the companion gives. The three others parameters (the companion's name, his or her picture and introductory message) are less relevant to the teaching strategy but are necessary for displaying the companion to students.

4 Interaction Between the Software and the Students

The objective of AMICO is to encourage students to give explanations in different ways and develop connections between different modes of representations. We wanted to make it as easy as possible for the student to input answers whatever representation mode they choose. In addition, the interventions of the companions are not fixed but dynamically generated during the session according to the actions of the students and the character of the companion set by the teacher (or the student).

4.1 Students' Modes of Expression

In algebra, students answer in mathural language or with algebraic expressions. This led designers to choose between: providing students with an editor for algebraic expressions or providing students with a list of propositions to compose his or her answer. In AMICO we chose the second option. Its main advantage is of course that it is easy to implement and, for this reason, many systems use it, in standard software as in research software (for example, [7]). Moreover, students are often not clever with keyboard input and this solution facilitates their work. But numerous educational critiques have been made of this solution: if the only activities proposed to the students are choices in lists of predefined answers, students are not confronted with the task of producing and articulating their answer, which is an essential competence to be acquired. Our observations in the classrooms however showed that students' activity with software is not restricted to interaction with interface: some use a draft or a calculator, discuss with peers or with the teacher. In the scenario of use that we consider for AMICO, students are working partly with the software but also partly without it. The interaction with the software gives them the opportunity to synthesize the results of these various activities. In this context, forcing the students to choose a type of justification and its expression from a predefined list helps to generate a socio-cognitive conflict: the student is confronted by different arguments and must give his or her position with regard to propositions.
AMICO provides to students two types of input: choose a justification from a list or compose a sentence from predefined pieces. On the one hand a student giving an answer, sometimes has to justify it. In this case, AMICO offers the student a number of answers in several combo boxes, one corresponding to each representation mode. With respect to our educational objectives, this method is a compromise between easy self-expression and pedagogical constraint. The combo box lists offer students' answers that we collected earlier and that are not always correct. They are intended to approximate a student's articulation better than articulations of teachers or books. On

the other hand, when one of the actors asks another actor for an explanation, the latter explains its reasoning process. As in [5;7], AMICO helps learners to write an explanation by providing sentence fragments. The list of fragments offered is made up dynamically according to students' previous choices. The companions' sentences are constructed from fragments in the same way because the companion's and students' role when interacting with AMICO are symmetric.
The effectiveness of this way of constructing explanations remains to be proved but first tests with students seem to raise no difficulties. However some teachers would prefer fewer choices. They ask for only one option chosen by us as prototype of students' answer for each type collected from our a priori educational analysis [9]. One teacher would prefer more academic formulations and opposes the design choice of displaying incomplete or wrong answers in the combo boxes. These reactions suggest to us that design choices can not be theoretically validated in isolation from teachers' conception of algebra education and their way of managing their classes.

4.2 Dynamic Evolution of Interactions With Companions

[13] proposed that educational agents must be autonomous i.e. they must be capable of pursuing goals in a complex environment and of adapting their behavior. The companions that we have designed in AMICO have very specific goal (disrupt the student or answer correctly or answer wrongly, or answer in algebraic language or in "mathural" language). In addition, each companion is able to adjust his or her goal. Companion's answers are not predefined but each interaction is calculated according to the step, the student's and companion's previous answers and the companion's goals. The companion will not answer the same way for two identical student answers. This is the reason we say that the interactions evolve dynamically.

5 Scenarios of Interaction

To create the scenarios of interaction, we began by finding types of exercises that allow discussion. We created two types of interaction: scenario of competition and cooperation. To test our ideas we looked for exercises in algebra that are suitable for discussion. In the prototype presented here, we choose three types of exercises:

- Deciding whether algebraic expressions are equivalent and justifying the answer
- Deciding whether a number is a solution of an equation and justifying the answer
- Comparing perimeters of polygons and justifying the answer

For each exercise, our a priori educational analysis identified a list of competencies involved in solving it. The educational analysis also identified a list of possible strategies.
Based on this educational work, we considered different scenarios. [12] proposed that the student can choose to solve the problem alone or in collaboration with the companion. For our part we invented two scenarios of interaction between the students (by dyad if possible) and the companion. The first scenario is founded on competition between the group of students and the companion, whereas the second is

founded on their cooperation. In the competition scenario actors (students or companion) are required to justify their answers. For each exercise, AMICO asks the actors to answer "yes" or "no", and then to choose a justification in several combos boxes according to their expression mode. During the first user test we made in our lab, students said they were not feeling the companion's presence ("We haven't the feeling that the companion is here"). So we try to specify a more interactive scenario. In the cooperation scenario, the two actors can ask the other for an explanation. To improve the feeling of presence of the companion, we introduced the first names into the dialogue. When the two scenarios are available for the same exercise, we let the students choose the scenario. In all the cases, at the end of the exercise, the mathematician institutionalizes the knowledge.
The *competition scenario* is based on a three steps exchange: the companion and the group of learners take turns three times in all. The objective of this type of problem is, from a pedagogical point of view, to show the rules of formation of algebraic expressions the students have constructed, and the role they assign to + and * signs, to parentheses and to exponents.
In the *scenario of cooperation*, the two actors (the group and the companion) cooperate to construct what they think is the best solution. This cooperation scenario only applies to problems that can be decomposed in two sub-problems. For example, an exercise presents two polygons including annotations carrying on the lengths of the sides and the question "Do the two polygons have the same perimeter?" The two sub-problems are in this case: "What is the perimeter of the first figure?" "What is the perimeter of the second figure?". The cooperative resolution of such a problem in AMICO has three steps. The first two steps require solving, after a discussion, the two sub-problems: find the perimeter of each figure. If the two actors agree on the solution (they agree on the two perimeters), they move to the third step (to say if the two perimeters are equal).

6 Testing AMICO with Students and Teachers

In the early tests we organized with the students we wondered if they would understand the trace of the dialogue generated by the system. We were concerned the dialogue was a little rigid and over-structured. These tests however did not raise difficulty in understanding the dialogue between the students and the companion. To the question "What do you think about companions", some answered "they are funny", "they are like us, sometimes they also make mistakes","he tried to disrupt us". To the question "Give strong points of the software", some answered "It is good because we aren't obliged to give the correct answers" "It is good because we can work with a companion and we aren't alone." Recall that in our design approach, these tests are not intended to establish a validation of the software in term of students' learning gains. This would require taking into account several other factors, for example the teachers' conceptions of the algebra education, their class management, the student's cognitive profiles and previous exposure to algebra teaching. The objective of these tests is to give design ideas, to test usability and to validate design choices.

Once the software was implemented, we tested it with five teachers. They were all interested in the idea of a virtual companion and in the fact that every companion has a different character. They thought that would please to students and motivate them. They appreciated that students and companions have several ways of expressing justifications and the diversity in the interaction, which, in their opinion would be beneficial for the students learning. They said that working on prototypic justifications and on various forms of expression would encourage students to question their conceptions. The teachers foresaw two types of uses in the classroom: a use with the whole class with a data show projector to initiate a discussion before a lesson about calculation with letters. This will allow the teacher to estimate the prior knowledge of the students in the class. The second is a use in individualised help sessions to elicit discussion in dyads and to challenge the students' misconceptions.
According to the teachers, one of the strong points of AMICO is the scope for them to generate other exercises and to modify the answer lists used to construct students' and companions' explanations. One teacher wanted companions give only correct answers, more in conformity with the answers that a teacher would give. The four others mentioned the rigidity of the dialogue and suggested some improvements. They were not in complete agreement with the mathematician's answer when proving that the equality of two expressions is false: the mathematician gives a counter-example. Three on five considered that this answer is insufficient and that it would be necessary to supplement it with an explanation: either a commentary in natural language on the correct rule, or by its demonstration. Those comments enlighten that the institutionalization step includes two parts: the answer and a meta-mathematic indication explaining the answer

7 Conclusion

AMICO has been implemented in Java using a UML methodology to develop its architecture. It is based on educational research. This software offers two different scenarios of interaction with virtual companions: competition and cooperation. We implemented four characters for the companions. These various characteristic allow adaptation to different learning styles and teaching strategies. The main limitation of the prototype in the current state is the rigidity of the automatically generated dialogue. A possible solution would be to introduce an agent to lead the dialogue who could encourage a better fluidity of interaction. In the longer term, we could develop a model of richer dialogue by studying the dyads' dialogues while AMICO's use. In the current prototype exercises are defined in text files and a teacher interface would be necessary to allow teachers to input new exercises without the help of a designer.
In our opinion, the version of AMICO we described here presents a triple interest. At the computer science level, AMICO implement a generic model that allows interactions to be automatically generated from file text specifications. This approach allows to instrument participatory design: the early testing process informs the design choices but does not lead to a "spaghetti program" that is often criticized in rapid prototyping methods. At the educational level, this prototype supplies an environment to allow students to work on various types of justifications and various forms of

expression. AMICO does not replace the teacher but can help as a catalyst in learning situations the teachers has set up. Finally, AMICO allows educational researchers to parameterize the companion and thus supplies them with a tool to study how students make sense with the different types of justifications.

Acknwledgements

We would like to thank Geoff Cumming who provided valuable insights and help us to improve the English version of this paper.

References

1. Aimeur, E., Dufort, H., Leibu, D., Frasson, C., Some justifications about the learning by disturbing strategy, AI-ED 97, Japan (1997) 119-126
2. Artigue, M., Aussadez, T., Grugeon, B., Lenfant, A., Teaching and learning algebra: Approaching complexity through complementary perspectives, 12th ICMI study, H. Chick, K. Stacey, J &J. Vincent, (Eds), Melbourne (Australia), vol 1(2001) 21-32
3. Bruillard, E., Delozanne, E., Leroux, P., Delannoy, P., Dubourg, X., Jacoboni , P., Lehuen, J., Luzzati, D., Teutsch, P., Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM, Revue Sciences et Techniques Educatives, hommage à Martiel Vivet, Hermès, vol 7, n° 1, (2000) 87-145
4. Chan, T.-W., Some learning systems: an overview, Innovate adult learning with innovate technologies, North-Holland, Collis B. Davies G. (1995) 101-122
5. Craig, S. D., Gholson, B., Garzon, M. H., Hu, X., Marks, W., Weimer-Hastings, P., Lu, Z., Auto tutor and Otto Tudor, AI-ED' 99 Workshop on Animated and Personnified Pedagogical Agents, Le Mans (1999) 25-30
6. Grugeon, B., Etude des rapports institutionnels et des rapports personnels des élèves à l'algèbre élémentaire dans la transition entre deux cycles d'enseignement : BEP et Première, Thèse de doctorat, Université Paris VII (1995)
7. Heffernan, N. T., Koedinger, K. R., Intelligent tutoring systems are missing the tutor: building a more strategic dialog-based tutor, Workshop "Learning algebra with the computer", ITS 2000, Montréal (2000)
8. Hietala, P., Niemirepo, T., The competence of learning companion agents, International Journal of Artificial Intelligence in Education, vol 9, n° 3-4 (1998) 178-192
9. Jean, S., Delozanne, E., Jacoboni, P., Grugeon, B., A diagnostic based on a qualitative model of competence in elemantary algebra, in S. Lajoie, M. Vivet, AI&ED'99, IOS Press, Amsterdam, , Le Mans (1999) 491-498
10. Les, J., Cumming, G., Finch, S., Agents systems for diversity in human learning, in S. Lajoie, M. Vivet AI&ED'99, IOS Press, Amsterdam, Le Mans (1999) 13-20
11. Ramirez Uresti, J. A., Teaching a learning companion, The Tenth White house papers. Graduate research in the cognitive and computing sciences at Sussex, Brighton, Eds. J. Halloran & F. Retkowsky (1998)
12. Ramirez Uresti, J. A., LECORBA : A learning companion system for binary boolean algebra, AI-ED' 99 Workshop on Animated and Personnified Pedagogical Agents, Le Mans (1999) 56-61
13. Shaw, E., Ganeshan, R., Johnson W. L., Miller, D., Building a case agent-assisted learning as a catalyst for curriculum reform in medical education, AI-ED' 99 Workshop on Animated and Personnified Pedagogical Agents, Le Mans (1999) 70-79