Humanistic Mathematics Network Journal

Issue 14 Article 16

11-1-1996

On the Use of Intelligent Tutoring Systems for Teaching and Learning Mathematics

M.D. C. Mendes
Institute of Mathematics and Statistics, University of São Paulo

M.G. V. Nunes
Institute of Mathematics and Statistics, University of São Paulo

C. A. Andreucci
Institute of Mathematics and Statistics, University of São Paulo

On the Use of Intelligent Tutoring Systems for Teaching and Learning Mathematics

M. D. C. Mendes, Department of Mathematics
M. G. V. Nunes, Department of Computer Science
C. A. Andreucci, Department of Mathematics
ICMSG University of Sao Paulo

INTRODUCTION

The idea of using computers in education is far from new. However, the more naive attempts have not been considering all the issues involved in such a complex task. As with any alternative tool, the use of computers for educational purposes demands caution in order to reach its goals; otherwise such effort can result in negative outcomes only. While technological advances continuously bring new design alternatives, conceptual problems which arise from the peculiarities of this medium seem to be frequently dismissed by computer scientists. In fact, no one can guarantee the tutorial system effectiveness (i.e. the student learning efficacy) only by virtue of its technological state. Students and human tutors have particular relationships with computers and this fact cannot be ignored during the design of tutorial systems. The real system's educational role strongly depends upon the roles of all other environmental components.

Computers cannot be seen as a panacea for educational problems. Some enthusiastics in education and computing areas tend to see technologies as the solution to most educational problems. Indeed, educators should not transfer the task of building an efficient automatic tutor to programers and computer scientists under the risk, among more serious repercussions, of the undermining of their own roles in the educational process.

First of all it is imperative to precisely determine what should be done by a computational assistant and what should be left to the human tutor. Moreover, the way in which the system is intended to reach its goals must be carefully designed. Only then, through a controlled experiment with students and human tutors, could the computer tasks be judged with regard to their learning goals.

This paper discusses some issues related to the benefits of tutoring systems and the care needed in the domain of mathematics. First, we will argue for a realistic learning environment where tutorial systems could yeld positive results. The human tutors' and students' roles will be also discussed. Second, we will list and discuss some relevant and problematic tutoring systems features. We conclude this paper by addressing some guidelines related to intelligent tutoring systems design in the domain of mathematics.

INTELLIGENT TUTORING SYSTEMS AND THEIR ROLE IN EDUCATION

Intelligent Tutoring Systems (ITSs) are education-purpose computer programs that incorporate techniques from the Artificial Intelligence (AI) community. They date back to the early 1970s and derive from CAI (Computer Aided Instruction) programs and differ from the latter in the use of AI tools in order to know what they teach, who they teach, and how to teach. The use of AI techniques presupposes an intention of producing human-based "good teaching," since most AI systems try to simulate human activities. Indeed, many ITSs are supposed to replicate all the teacher's activities. ITSs should neither be of naive Skinnerian type linear programs nor completely take over from the teacher. One useful role for ITSs ties in their potential of working as intelligent tutoring assistants. In this framework, AI techniques are welcome and necessary as well.

The need for better quality teaching and for more effective results has always been publicized. Teaching is a very complex task which demands knowledge, ability, mature thought, intuition, self-confidence, empathy, capacities of seeing and hearing, and the capacity of motivating the students, among other human abilities.

Teaching is a special task since it involves the sharing of human responsibilities in society. Children, youngsters, and adults should have their individual natures taken into account in this process. The teacher's behavior must then be adjusted to each student. Learning, on the other hand, is exclusive to the student, and no one, least of all, the teacher can take over in that process. The teacher's task is to provide for the student's learning by creating good external conditions for the development of the learning capacity. Learning is then a subjective process and depends on personal experiences. Two circumstances will determine its adequacy. The first is the motivation to study the subject and overcome knowledge difficulties. The second is the promotion of a safe environment for the student in which he/she gets more independence by overcoming his/her own reasoning and knowledge limits.

Not being a substitute for the teacher, an ITS is a teaching support tool, fitted to the necessities of revision, diversification, flexibility, problem solving, progress in content, etc. Moreover, in the classroom, while the teacher's pace of presentation depends on his/her own experience, through an ITS, the student can determine the pace at which the knowledge should be presented.

As a computational assistant, an ITS would complement teaching activities which are not covered by the teacher. ITS would be stimulating as long as it can be different from the traditional classroom model. However, three main issues can endanger the function of a computational assistant: its limited capacity for expansion, its set of teaching methods, and its inability to understand students' idiosyncrasies. These limitations, nevertheless, can stimulate new questions, analogies and corelations which are unusual in traditional settings. These questions can thus play a role enhancing discussion inside the classroom.

The individual interaction with an ITS favors the student's identification of his/her own mistakes -- a challenge that could imitate a game-like interplay with the machine. Moreover, the ITS can provide the teacher with help in the learning by doing approach which is so difficult to implement in classrooms. This environment also favors the development of intuitive reasoning such as the forecast of right answers. What will be the result? What will be the way to reach it? These questions will drive the procedures even if a realistic student-system dialogue is impossible. The sucess of the use of intuitive reasoning demands the use of analytical reasoning, which depends on the possibility of succeeding and so on.

The teacher's role in such an environment mutates as long as the students are more participatory, offering the former opportunities to discuss concepts outside the realm of the ITS. The teacher is also supposed to indicate why, when and how much the computational assistant should be used. The students' productivity offers the teacher parameters for the system feedback.

LEARNING MATHEMATICS THROUGH INTELLIGENT TUTORING SYSTEMS

One of the most insistent problems in mathematics education is the aversion that many students feel towards this subject. Even students at graduate school levels in mathematics or computing courses often have problems related to the disciplines involving some concepts they are supposed to already be acquainted with. The literature has many studies concerning errors made by students and the persistence of misunderstandings of such errors. There are also other studies reporting high rates of failure among students in mathematics. This probably can be attributed to their experiences in learning mathematics. The use of strategies which minimize rote repetition of algorithms would be of much value [1]. The repetition approach probably leads the students to construct an improper schema to solve the problems by themselves. Such an improper approach is reinforced by doing a large list of similar exercises with the same interpretation.

This is an important point that has to be thought of attentively. The lack of understanding of a concept may not be due to the concept itself. It is often due to an insurmountable barrier for the student which is not the current concept, but a previous one which is a prerequisite to that in question.

As teachers we cannot forget that before introducing a new concept to the class we must have it clear in our mind what adjacent ideas are also involved. For example, the lack of understanding quantification is often a barrier for students in developing a more sophisticated understanding of limits and continuity. This could explain, for example, why students fail to understand calculus and a really long list of other topics.

This example illustrates the necessity for students to be able to express the prerequisites of the concepts they are supposed to learn. It therefore seems that finding information about the idiosyncratic learning methods of understanding concepts we are going to teach, how they are learned, and what we as teachers can do to enhance the student's logical thought might contribute to the goal of improving the students' understanding of advanced mathematical concepts.

We believe that an effective understanding of a mathematical concept depends on individual efforts to construct these ideas by the students themselves. And it is possible to detect, through research, the different ways in which this can take place. We also believe that it is possible to develop computer-implemented tutors which are designed in order to stimulate the constructions detected by the research, towards a reasonable acquisition of mathematical concepts. It is important to notice, however, that a mathematician has his/her own understanding of the involved concepts and it is up to the teacher to have the awareness to avoid the bias of that understanding when the analysis of students' styles of learning is made. It is true that it is not that easy to completely avoid this (although implicit) interference; however, an effort should be made to minimize this as much as possible.

Dubinsky [1] pointed out that it is important to observe that any description of the concept must not only be "mathematically wrong" or "mathematically correct" but must also embody all of the subtleties and other styles used to understand the subject. We are sure that all of these variables come to enrich the process of analysis of the possible ways of learning, giving us many ramifications of the concept in question, reflecting its varying role in the full spectrum of mathematical endeavours.

Of course there are several ways to describe a mathematical concept. The process of its acquisition can be determined by observing students in the process of construction of the concept. The students' successes and failures can be important clues to the essence of the ongoing learning process. An accurate analysis of these components can reveal the defective points that lead a student to make mistakes, which if appropriately explored, would certainly contribute to the main goal teachers must have: to enhance the student's performance as a problem solver.

As Dugdale pointed out [2], presently we have the possibility of using AI methodologies for the realization of expert systems, which permit the use of computers to be extended to fields that some years ago only human experts could master. One such field which could paricularly gain from this is mathematics. We do not refer here to those systems which provide a one-way teaching interaction, but those which have a mixed-initiative teaching dialogue, which is individualized to the needs of the student as an individual. In this way, the analysis and the diagnosis processes must be present as one of the main factors. The intelligent tutoring system used to help students in learning mathematics is supposed to act as an assistant to the teacher. Its task is to support both student and teacher in the teaching-learing r elationship.

Thus, it is a matter of great weight to have a cooperative environment to help students in learning new concepts and prerequisites as well. It is important to emphasize that the ITS must lead the students to dominate their own problems step by step, encouraging them to become active, creative, and independent learners. The ITS system may also allow the student to choose a better way for himself/herself, resulting in a rich environment for exploration. We believe that learners will become more and more motivated and confident; they can find out that the more they learn the more they are able to do.

DISCUSSION

The questions that arise are if and how computational assistance can help in teaching mathematics. The prerequisite barrier can be overcome by the modelling of the students' knowledge by the system. But this is not quite simple. The nature of the students' knowledge to be considered and the rules to manage it are still major problems of ITS design. Most ITSs use poor measures of students' knowledge such as numbers for category levels and quantity of right and wrong exercises. More qualitative measures such as the students' knowledge about the relationship between concepts ought to be taken into account. An ideal student model should be made up of information about the history of the student-system dialogue, as well as information about the student's performance during problem solving. In terms of knowledge representation formalisms, AI-based models combine a frame-based schema with production rules and an inference mechanism for deriving new information about the student. However, the type of each information set and the rules connecting them are far from simple to provide.

However, it is also not simple to detect students' misconceptions. The cause for the students' errors can rarely be localized to a unique concept. Indeed, the method of relating concepts may be the problem focus. The reasoning method is supposed to supply the relationship between the concepts that the student detects in that domain. While the computational artifact seems to be adequate, the qualitative nature of the information remains open for further research investigation.

Probably the biggest problem in designing tutorial systems in the domain of mathematics is the need to handle reasoning. Beyond concepts, the student is supposed to learn the underlying reasoning. Therefore, handling the reasoning requires from the system a description or formalization of the knowledge. The computer should stimulate the student's reasoning, while deeper discussions should take place among classmates and teachers. Once this is done, another issue remains unsolved, which is the importance of stimulating the student to develop his/her own method of reasoning. A useful intelligent assistant should be able to understand and classify that method, or even learn a new one. However, students have idiosyncratic methods of solving problems and even sophisticated systems which know several methods cannot handle all the existing possibilities [4]. While human teachers are able to learn the students' methods through dialogues with them, the use of machine learning -- based approaches is in its early stages [3].

The computer should stimulate the student's reasoning, while deeper discussions should take place among classmates and teachers.

The imposition of the teacher's way of reasoning can be avoided through the use of different solution methods appropriately presented. Since it is not possible to cover all styles of human thinking, we can start by associating the methods with the concepts in order to better present them to the student. However, only the explicit representation of this knowledge within the system can guarantee its capacity of detecting students' misconceptions and explaining its tutorial strategies. A still open problem related to this is the need for a dialogue where the student can explain his/her way of reasoning. Not only is this useful to enhance the system's knowledge about the student, but is also crucial for the student to become conscious of his/her own failures and successes. Here one should bear in mind the limitations imposed by the computer interface dialogues, especially natural language-based ones -- which still do not allow for a cooperative dialogue with the student. In a cooperative environment, the more active the participants are in the discussion, the more productive and effective the learning process is. The computer should stimulate the student's reasoning, while deeper discussions should take place among classmates and teachers.

As pointed out above, the computer should be part of an environment together with the students and the teacher. As such it is not completely true that the student is the only ITS user. ITS should be able to interact with teachers and students separately as special and equally important users. The role of the teacher as an ITS user must involve two issues: the system validation and the teacher's evaluation of the student.

By system validation we mean the access to the system knowledge bases (domain, student, tutorial) and to the rules that control them during a special session targeted to the teacher. As an expert for domain and/or tutorial knowledge, the teacher should interrogate the system in order to get a system radiography. The underlying assumption is that, as a dynamic tool, an ITS should be constantly adjusted, improved, and corrected.

The evaluation of the student takes place during or right after a student session in order to obtain information about the student performance. This data includes the student model information and the system's justifications for its decisions. System justification has not been granted enough attention in ITS projects. We cannot forget, however, that one of the most important features -- indeed requisites -- for an 'intelligent' system is its capacity for explaining or talking about itself. More important than the adequacy of its designation, this feature gives confidence to its users, the lack of which can jeopardize the entire learning process.

CONCLUSIONS

A useful computational assistant should know many presentation methods and know where and when they should or should not be used. This 'intelligent' feature is mandatory in any ITS and can be supplied by computer resources. In spite of this, there are some useful guidelines which should be followed to achieve successful learning. In mathematics, in particular, software must be attractive and challenging. This does not mean that it always must be camouflaged in games or the like. We do believe that with the cooperation of students and teachers, and only then, will it be possible to design useful assistant mathematical softwares.

Idiosyncratic learning methods demand different system characteristics. For example, some students prefer to be constantly evaluated, while others would prefer more complex evaluation methods; others like to know the system's teaching methods, while others would prefer not to see the system as a teacher. When and how the internal system knowledge should be presented can be a question of preference to the student, but it is mandatory for the teacher who must have access to the system in order to check its behavior. So the teacher should point out what system information she or he would like to access and how this information should be presented .

Students, on the other hand, play a very special role in the tutoring system design. In addition to expressing their preferences, students can determine the system's success or failure, for they really can say what and how they have learned. While learning can be difficult to measure, it is easy to preview that learning is almost impossible to achieve when the students are left out of the decision process. The ways through which the student should participate remain to be further investigated. Cognitive aspects must be taken into account in order to detect the students' idiosyncratic methods of reasoning.

Based on the above ideas, we have designed and implemented a system prototype aimed to support elementary school students in learning plane geometry. The system TEGRAM provides a set of activities based on Tangram. The activities include measure and shapes of plane figures and similarity, among others. The student can use the system according to his/her own cognitive level. The system tries to evaluate the user through a student model and proposes a new set of activities made adequate to the detected level. Initial results point to positive student reactions. The underlying approach is to allow the student to choose his/her way to solve the problems, which is what makes the system quite challenging. However, this freedom does not prevent the system from suggesting an appropriate sequence of activities for the student, based on some knowledge about his/her performance. We are on the point of reiterating that the process of learning and teaching mathematics has much to gain from the use of an intelligent tutoring system as an assistant.

REFERENCES

[1] Dubinsky, E. "Teaching Mathematical Induction I." *The Journal of Mathematical Behavior II*, 1991.

[2] Dugdale, S. "The Design of Computer-Based Mathematics Instruction." *Computer Assisted Instruction and Intelligent Tutoring Systems -- Shared Goals and Complementary Approaches* (eds. J.H. Larkin & R.W. Chabay). Lawrence Erlbaum Associates, Hillsdale, New Jersey. 1992, pp. 11 - 46.

[3] Gilmore, D. and J. Self. "The application of machine learning to intelligent tutoring systems." *Artificial Intelligence and Human Learning -- Intelligent Computer- Aided Instruction* (ed. J. Self). Chapman and Hall Computing, 1988, pp. 179 - 196.

[4] Ridgway, J. "Of course ICAI is impossible...worse though, it might be seditious." *Artificial Intelligence and Human Learning -- Intelligent Computer-Aided Instruction* (ed. J. Self). Chapman and Hall Computing, 1988, pp. 28 - 48.