A Didactics-Driven Intelligent Tutoring System

Alfred Errera, Lambda, Israel
Dorit Patkin, Kibbutzim College of Education, Israel
Tzachi Milgrom, Science Teaching Center, The Hebrew University of Jerusalem, Israel

Abstract

Most Intelligent Tutoring Systems (ITS) have remained research tools and have not succeeded in bridging the gap from the laboratory to the school. The Guided Autonomous Learning System (GALS) is a new ITS model that is grounded in a substantiated model of learning, that would provide diagnosis and adapted remediation of errors, while at the same time paying attention to environmental factors and User-Machine Interface. GALS has been applied to a primary school mathematics program that is currently in use in Israeli classrooms. The paper presents the principles underlying the development of the GALS model, and describes the results of its experimental implementation in Grade 1 classes.

Introduction

When the concept of Intelligent Tutoring Systems (ITS) was first proposed, great hopes were raised that efficient models would be developed and implemented in schools (Murray, 1999). In fact, however, only a few such systems have transcended the confines of research models and have actually been implemented in classroom settings (Urban-Lurain, 1995). According to Rosenberg, two main reasons explain the lack of use of ITS in actual education:

1) “The systems are not grounded in a substantiated model of learning”.
2) “Testing is incomplete, inconclusive, and in some cases totally lacking” (Rosenberg, 1987).

Although it is clear that a good ITS must be able to handle student errors, Rosenberg points to the models by Burton and Anderson (Burton, 1982 and Anderson, et. al 1985) and asserts that “an exhaustive enumeration of error types is not a model” (Rosenberg, 1987, p. 8). He also argues that “lists of error types are only part of the raw data for a model of tutoring. Unanalysed listings do not illuminate the tutoring process.”

The following paper presents a model of a new ITS, that is indeed grounded in a model of learning, and includes diagnosis and remediation tools. This didactically-based ITS has been implemented in the study of mathematics in primary schools in Israel, and the results of its classroom use will be described. At the time of this writing, the GALS ITS is being used by more than 2,000 Grade 1 students.

The Guided Autonomous Learning System (GALS) ITS Model

As early as 1973, Brown noted the positive synergism that could result from combining the power of the new computers and advances in computer languages with research in didactics (Brown, 1973). For this reason, the new GALS ITS uses object-oriented programming and multimedia resources and we incorporated findings from didactic research in its design.

Anderson asserts that “cognitivism does not imply outright rejection of decomposition and decontextualization” (Anderson, 2000). The GALS model applies two basic principles stressed by Anderson:

1. “Assessing learning and improving learning methods requires careful task analysis at the level of component skills, intimately combined with study of the interaction of these skills in the context of broader tasks and environments” (p.3).
2. “Assessing learning and improving learning methods requires research and instruction in contexts that are consistent with the scopes of the skills currently under investigation.” (p. 4).

The GALS mathematics curriculum for first grade is comprised of more than 200 activities, each of which deals with one central concept. These activities are arranged in ascending order of difficulty, from the most simple to the most complex. The GALS model was first applied to mathematics rather than to other disciplines because the structure of mathematics enables the development of a logical program structure (Suppes, 1967).

Open models, such as simulations, play a very important role in the learning process; however, it is very difficult to deal with diagnosis and remediation in an open environment. For this reason, the GALS model, in which every activity has a well-defined goal, includes the following types of activities:

1) “Open-ended” activities, i.e. activities that can be approached in many ways and have more than one correct answer;
2) “Semi-open” activities, i.e. activities that have a goal, but which allow students to work in the way that they want;
3) “Closed” activities, i.e. activities that have a goal and only one way to reach it.

The inclusion of open-ended, semi-open, and closed activities in the GALS model enabled the integration of powerful diagnosis and remediation tools.

Each activity in the GALS model begins with an explanation that teaches the students a basic concept for the first time or reminds them of a concept that they have learned previously. The computer then demonstrates how the interface for the specific activity works (e.g. “just click,” “draw,” “select,” etc.).

Following the demonstration, the student is asked to solve the identical exercise that the computer solved. Contrary to what one might expect, we did not find that students in such situations simply imitate without understanding; rather, it was found students who were not truly ripe for an exercise were not necessarily able to repeat the demo.

An explanation for this can be found in the assertion in Vergnaud's book on Vygotsky: "We can only imitate what lays in the zone of our own intellectual capacities" (Vergnaud, 2000, p. 28).
After the student solves the first exercise, s/he is presented with 2-7 similar exercises. These exercises are generated by the computer pseudo-randomly, such that two different students working on the same activity simultaneously receive different exercises, and students who do the same activity twice encounter different exercises. If the student solves an exercise correctly, s/he receives a funny animation, gets a green light, is awarded points, and is automatically presented with the next exercise.

Behaviorist models, which consider students to be a "black box," resulted in major failures (Hativa, 1988). For this reason, any new ITS models should integrate an analysis of frequent errors and misconceptions, and should be able to deal with the reasons that underlie those mistakes. In addition, new ITS models should provide assistance and remediation built on mental models.

The strength of the GALS model is revealed when a student makes a mistake. For each concept presented in the program, the developers reviewed common mistakes and determined prevalent misconceptions. They then built state-machines that diagnose each foreseeable mistake. Findings of research regarding errors in specific domains were incorporated when available (e.g. Brown, 1978; Van Lehn 1982; Gray, 1994), and when there was no existing body of findings, research was conducted in order to map common errors.
As mentioned previously, it is not sufficient to know what the frequent errors are; rather, it is necessary to have a plan of action for instances in which errors of this nature are encountered. For example, the mistake most commonly found in response to the exercise "3 + __ = 8" is the answer "11", i.e. the student adds 3 + 8 instead of determining the missing number (Conne, 1988**)**. What should be done when a student makes such a mistake?
Cox has noted the need for an appropriate teaching method for providing remediation for specific errors: "Research on what teaching methods are appropriate for remediating specific errors are almost inexistent" (Cox, 1975).
Two basic approaches to correcting specific errors can be described:
a. The Piagetian "accommodation-assimilation" model: In simple words, if a learner makes a mistake in a given problem, he must abandon the algorithm applied and construct a new one that better fits the situation. This process is called "assimilation." If the new algorithm works, the learner records the match between the algorithm and the situation, and correctly applies the algorithm upon encountering similar situations in the future. This is known as "assimilation" (Siegler, 1991). If we apply this model to our case, the appropriate remediation would be to inform the student that he made a mistake (e.g. by saying "you made a mistake," "you're almost there," or "try again") and to then allow him to build a new algorithm and to assimilate it.

b. The “private tutor” model: According to Anderson, a private tutor who is faced with a student who makes a basic mistake will commonly respond by simply teaching the concept again.
The basic problem with incorporating the Piagetian model is that many students are not bothered by the fact that their algorithm does not work when applied to a specific problem; moreover, even if they are bothered, they often are not capable of constructing a new algorithm independently. The “private tutor” model of remediation is also problematic, in that students do not build their knowledge base by themselves if they are simply taught concepts again. Consequently, the GALS model developed an alternative method of responding to errors**,** which falls between the two models described**.**
Using Vygostky’s principle of a “zone of proximal development” (Vygotsky, 1986), the student’s mistake can bee seen as an indication that s/he is having difficulty “climbing the step” that leads up to the new concept. In such situations, students should be given a “stool” that is appropriate to their specific problem, so that they can “climb the step” by themselves. In our design of the GALS model, these “stools” took the form of “hints” that were carefully constructed based upon an analysis of the causes of common mistakes and misconceptions, and that give a basic adapted remediation.

Thus, for example, in our case of the mistaken response to the equation 3 +__= 8, most students who had difficulty did not understand the order of the equation. They saw the 3, the plus sign, the 8, and the equal sign, and they simply solved the equation 3 + 8 =, perhaps wondering if the teacher who wrote the exercise was feeling quite right. In response to such a mistake, one simple but efficient hint could be: “3 plus 11 does not equal 8” (which points out that student did not arrive at the correct answer). Similarly, the hint could be phrased “3 plus what equals 8?” (which explains the order of the equation). After receiving the hint, the student could then be given a second chance to solve the exercise.
If the student fails again after receiving a hint, GALS implements a method close to the “private tutor” model and presents the student with the solution, but only after explaining to the student what is wrong with the answer. (It is important to note in this context that since the exercises are generated in real-time, the hints and corrections are also generated in real-time by powerful engines.)
This method of remediation proved to be very efficient. For most students, the hint (and if necessary the full explanation) was sufficient to enable them to build a correct algorithm. However, occasionally it was found that students were unable to build a correct algorithm despite the fact that they received help and were shown how to solve this type of problem. As described below, the GALS model’s “Learning Manager” deals with such cases.

The Learning Manager

Lepper and Gurtner's presentation of the advantages of computers as personal tutors included the fact that computer programs can be individualized for learners with different skills and capabilities (Lepper and Gurtner, 1989). In the GALS model, at the beginning of the school year, each student begins the program with the same first activity (albeit with different generated exercises). If she succeeds in the activity, she gets the next one, and so on, following the order of the decomposition of the program in basic units, which each deal with one specific concept.

One might think that the students would all proceed through the program in a common, linear path; in fact, however, no two children ever followed the same course through the program. This is because if a student encountered difficulties in an activity, the "learning manager" would send him to a revision activity that would have him complete all the prerequisites for the activity in which he failed. This revision activity could either be an exercise that the student had completed previously or a special remedial activity that would only be encountered by students experiencing difficulty. Conversely, students whose performance was found to be very good are sent by the "learning manager" to special activities reserved for students with high levels of achievement.

Although in most cases, the learning manager and the expert systems for each activity enable students to bridge the knowledge gap and move ahead, occasionally a given student still does not master a specific concept. In such cases, the teacher receives a special report that lists the students who experienced pronounced difficulty, as well as the specific difficulties of each student. This enables the teacher to provide appropriate remediation that is focused on the specific problems of specific students, and even to directly access the computer activity that revealed the student's difficulty.

User-Machine Interface

According to Lepper and Gurtner's outline of the advantages of computers as personal tutors, the computer always remains patient, nonjudgmental, and supportive. Moreover, because the computer is always fair and impartial, it may minimize the pernicious effects of teacher prejudices or favoritism (Lepper and Gurtner, 1989). For this reason, effort was made to build a user-friendly interface, to create a program that is both patient and funny, to determine objective criteria of success, and to be sure that the interface includes a dynamic adaptation to each learner.

The Experiment and Implementation in Schools

In 1998, the Israeli Ministry of Education appointed a commission to supervise an experiment that has been conducted by an independent professional team. During the first year of this experiment, the GALS-based program was utilized in Grade 1. In each of the following years, participating students moved on to the next grade of the program, while new students began the program at each grade. At the present time, more than 800 students are participating in the experiment**.** In addition, at the start of

the 2000-2001 school year, the program became commercially available to Israeli schools, resulting in its current use by over 2,000 first grade students.
Beyond the computer program with hundreds of multi-media challenges, the computerized GALS-based math program in use in Israeli schools today is accompanied by a set of tools that includes student workbooks, ready-to-use exams, enrichment worksheets, manipulatives, and a teacher guide with suggestions for hundreds of discovery activities that can be done by groups or individuals. Students using the program spend an average of one class hour a week on the computer. The learning manager tracks the progress of each student, and teachers receive reports about the achievement of the class as a whole as well as the about the performance of individual students.

Results

The main results found by the independent professional team have been the following:

1. A comparative summative test of achievement of first grade students showed significantly better achievement ($p<0.001$) in the research group (10 classes, 217 students) than in the control group (10 classes, 211 students).

2. The integration of computer technology in the learning process was proven to contribute greatly to student achievement.

3. Most students mentioned the integration of the computer as the main factor of satisfaction of the program.

4. Both teachers and students reported a high level of satisfaction with all the parts of the program.

5. In the GALS program, students receive explanations each time they do not succeed in an activity. In this manner, beyond its role as a mechanism for practice and as a diagnostic tool, the program serves primarily as a teaching tool that enables the students to correct their mistakes and to understand the reason of their failure.

6. In many cases, it was found that the computerized activities presented students with concepts that had not yet been covered in class; nonetheless, the explanations provided by the program enabled the students to succeed in these activities. In addition, it was found that students who were exposed to concepts for the first time on the computer succeeded in the parallel textbook activities without needing additional explanations.

7. Students who encountered problems in a specific activity usually received efficient help and corrections, and were able to succeed and to move on to the next activity.

8. Most students succeeded in all the activities either after their first attempt or upon returning to the activities after being given additional remediation activities.

9. It was found that almost no random answers were given by the students.

10. Teachers reported that the software provided weak students with a learning experience that allowed them to remain "tuned in" during the lessons. This was not found to be the case with such students in previous years (Milgrom, 1999, Milgrom, 2000).

It should be noted that following the initial experiment, the Israeli Ministry of Education recommended pursuing a three-year trial and committed itself to continuing support for the program's development.

Conclusions

These results seem to indicate that an Intelligent Tutoring System grounded in a substantiated model of learning can efficiently be used in classrooms settings.
More data about the effectiveness of the GALS model are expected to be available at the summative evaluation stage of the ongoing experiment, and could be validated in the course of a new experiment that has begun this year with first grade classes in France.
The Guided Autonomous System (GAL) model could be applied to the study of mathematics in higher Grades, as well as to other subject matters.

References

Anderson, J. R. and Raiser, B.J. (1985), The Lisp Tutor, *Byte,* April 1985.

Anderson, J.R., Reder, L.M., Simon, H.A. (2000), Applications and Misapplications of Cognitive Psychology to Mathematics Education. http://act.psy.cmu.edu/personal/ja/misapplied.html

Brown, J.S. (1973), Steps Toward Automatic Theory Formation, *Proceedings of the International Joint Conference on Artificial Intelligence*.

Burton, R.R. (1982), Diagnosing Bugs in A Simple Procedural Skill, in Seaman, D. and Brown J.S. (eds), *Intelligent Tutoring Systems.* Orlando: Academic Press.

Conne, F. (1988), Comptage et écritures en lignes d'égalités numériques, *Recherche en Didactique des Mathématiques,* 9,1, 71-116.

Cox, L.S. (1975), Diagnosing and remediating systematic errors in addition and subtraction computations, *Arithmetic Teacher,* 2, 151-157.

Gray, E. (1994) Spectrums of Performance in two Digit Addition and Subtraction, *Proceedings of the Eighteenth International Conference for the Psychology of Mathematics Education,* 3, 1994.

Hativa, N. (1988), Sigal's Ineffective Computer-Based Practice of Arithmetic: A Case Study, *Journal for research in Mathematic Education,* 19, 3, 195-214.

Lepper, M. R. & Gurtner, Jean-Luc (1989). Children and computers: Approaching the twenty-first century. *American Psychologist*. 44/2, 170-178.

Milgrom, T. (1999) Evaluation report of the "Heshbon 10" kit for Grade 1. Israel Ministry of Education.

Milgrom, T. (2000) Evaluation report of the "Heshbon 10" kit for Grade 1 and 2. Israel Ministry of Education.

Murray, T. (1999), Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, 10, 98-129.

Rosenberg, R. (1987). A critical Analysis of Research on Intelligent Tutoring Systems. *Educational Technology,* 27(11), 7-13.

Siegler, R. (1991), Children's Thinking. Englewood Cliffs, NJ: Prentice-Hall.

Suppes, P. (1967), Some theoretical models for mathematics learning, *Journal of Research and Development in Education*, 1, 5-22.

Urban-Lurain (1995) An Historic Review in the Context of the Development of Artificial Intelligence and Educational Technology.
http://web.cps.msu.edu/~urban/ITS.htm

Van Lehn, K. (1982) Bugs Are Not Enough: Empirical Studies of Bugs, Impasses and Repairs in procedural Skills, *Journal of Mathematical Behavior,* 3,2: 3-71.

Vergnaud, G. (2000), Lev Vygotski, Pédagogue et Penseur de Notre temps, Hachette-Éducation.

Vygotsky, L.S. (1986), Thought and Language. Kozulin, A. (Ed.): The MIT Press, Cambridge, Mass.