Can We Teach Computers to Teach?

Eric Mazur

Computers have yet to cause the revolution in physics education that has long been expected

The computer has become a mandatory tool in academia and business. A walk around a university campus is likely to show that there are as many computers as there are students, faculty and staff. Outside the campus, many of our daily activities have to do with computers: banking, reservations, check-out registers at supermarkets, not to mention all the computer-generated mail we receive every day.

Surprisingly, in education, the computer is still a not-much-appreciated newcomer. One reason for this is that until not so long ago, computers were text-oriented, accepting only commands in the form of words. Such "educational" software usually emulated multiple-choice exams. Naturally, such programs could not keep anyone's attention for very long. Another reason for the small inroads made by computers in education is that they usually excel at doing routine tasks, while education is normally anything but routine. Good teaching requires that one constantly adapt oneself to the students.

Many are skeptical about the possibilities for computers in education. Similarly, many people, including myself, were skeptical about pocket calculators when they were first introduced in the early 1970s. People were going to become lazy and "forget" how to add, subtract, multiply and divide. In less than two decades, pocket calculators have indeed become ubiquitous and are used to do even the simplest additions. Of course, long before the introduction of pocket calculators, there was the abacus, which is still in use to this day. On a recent trip to China, I was paying for two items costing 10 yuan each, and much to my surprise (and amusement), the person in the store used an abacus to obtain the total price of 20 yuan.

The art of performing simple arithmetic in one's head may soon be lost. Does it really matter? It is surely

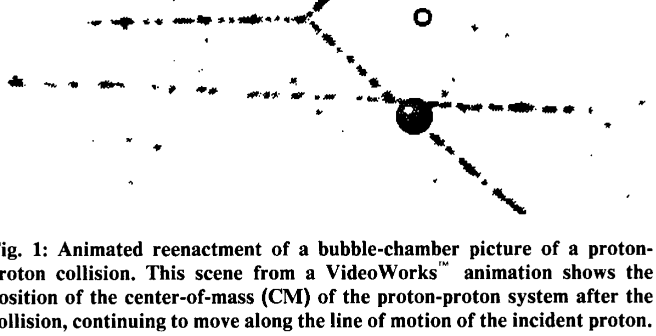

Fig. 1: Animated reenactment of a bubble-chamber picture of a proton-proton collision. This scene from a VideoWorks™ animation shows the position of the center-of-mass (CM) of the proton-proton system after the collision, continuing to move along the line of motion of the incident proton.

Eric Mazur is Professor of Physics and Gordon McKay Professor of Applied Physics at Harvard University. He divides his time between research in laser physics and teaching, and is interested in finding innovative ways to use personal computers in education. He is also the author of Essence of Physics, *which won first prize in the tutorial category of the First Annual* Computers in Physics *Education Software Contest.*

not a simple question, though many oppose such a development very strongly. On the other hand, not many people worry about the lost art of performing calculations using logarithm tables. The widespread availability of pocket calculators has made it possible to concentrate more on other aspects of calculations instead of looking up logarithms in tables. On a similar note, symbolic manipulation programs take away many of the routine aspects of mathematical problem-solving—perhaps in time, manual integration and differentiation will become obsolete too, together with handbooks of mathematical functions.

What about computers in education then, where very little routine exists? I don't think computers will replace teachers, but I am confident that computers will play an important role in improving teaching. To illustrate this, I will discuss several projects pertaining to computers in education in which I am involved.

My computer usage in the classroom coincided with the introduction of the graphic user interface offered with the Macintosh in 1984. In that same year, I started teaching an introductory physics course for science majors, and began using the computer for classroom demonstrations. Most of these were animations, created using a graphics animation program called *VideoWorks.*™ These animations have a dual purpose. One is to clarify certain concepts that are hard to visualize using static drawings. The other is to provide a break in the lecture, to keep the attention of the audience.

Let me give an example. To illustrate the concepts of center-of-mass and conservation of linear momentum, I project two *VideoWorks* animations onto a large screen in the lecture hall. The first one shows several collisions in one and two dimensions in slow motion. The collisions are then repeated, first showing the position of the center-of-mass and the direction of the impulsive forces. The second animation first shows a scanned bubble-chamber picture of a proton-proton collision, as printed in many textbooks. A proton, represented by a little sphere, appears from the side of the picture, follows the trace in the picture, and then collides with another proton originally at rest at the center of the picture. After the collision, both move away at a right angle, as is typical for an elastic collision.

So far, the animation just serves to explain what the students are looking at. The animation is then replayed with the position of the center-of-mass added, which moves at half the speed of the incident proton, and *continues* to move in a straight line after the collision (Fig. 1). The students can *see* the motion of the center-of-mass and the relation between the position of the two protons and their centers-of-mass. The next replay takes this another step further: the center-of-mass is pinned down, and the students can view the collision from the center-of-mass frame. The background now slides slowly to the left. The final replay shows the same view from the center-of-mass frame, but now without the background picture. This allows the students to observe that in this frame the two protons both approach the center-of-mass with equal speed, and that after the collision they still have equal but opposite velocities.

Over the years, I developed many such animations for a wide variety of topics. They were generally well-received by the students. An article I wrote in *Academic Computing*[1], on the use of computers in physics, prompted the physics editor of W.W. Norton (publisher of Ohanian's *Physics* textbook) to ask me if I was interested in developing software to accompany their textbook.

The resulting software package, *Essence of Physics*, a Hypercard program for the Apple Macintosh, is an attempt to make the personal computer part of teaching physics both inside and outside the classroom. Whereas most educational software is designed for a particular topic, this program is intended to be used throughout a one-year introductory physics course.

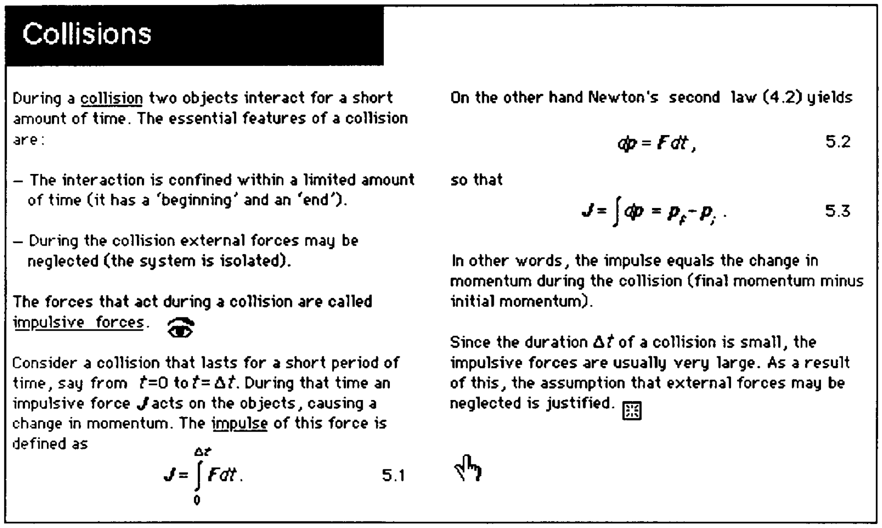

Fig. 2: Notecard on the subject of collisions from *Essence of Physics*. The user can click on any word to see its definition. The "eye" symbol on this card starts an animation that shows a simple two-dimensional collision between two spheres and the role that the impulsive forces play in the collision.

Essence of Physics

I started working on this project with the following design goals: It had to work on existing and already widely-available technology; and it had to be usable by students on their *own* computers, even without a hard drive. It could, therefore, not rely on any type of costly video technology. It had to be made available cheaply, and, as I mentioned before, it had to contain enough material to be useful during an entire one-year introductory course.

The program has three different features: (1) tutorial text for each of the 24 main subjects of a typical one-year introductory physics course; (2) interactive animations and demonstrations; and (3) interactively-solved problems. The student is in constant control of the program, and can jump at will from topic to topic, view some demonstrations, or solve a problem. The student interacts with the program by pointing and clicking the mouse or, if needed, by entering algebraic expressions or text. From the student's responses, the program tries to analyze the student's strong and weak points and provide specific feedback.

The tutorial text, which is presented in the form of notecards, is the frame that connects the other parts of the program. For each topic covered, e.g. forces and Newton's laws, collisions, electrostatics, special relativity, etc., there are between five and ten notecards (see Fig. 2). These notecards, which contain text from my own lecture notes, provide an overview of the minimum knowledge a student should have of each topic. They are compact, yet complete. In a sense, they reverse the current trend in textbooks, which, with an average of over one thousand pages, are now approaching encyclopedic proportions. The text in the program is meant for a quick review of the *essential* material. Generally, text is more easily read in print than on a computer screen, but the computer opens up possibilities, as described below, that do not exist with printed text.

At all times the student can *interact* with the text. A click on any word highlights that word and brings up its definition in the program. Clicking on the word "force" or "forces", for instance, will instantly call up the notecard where the concept of force is introduced. After reviewing this concept, the user can return to the original notecard with one click. All text is cross-referenced, so the student is not limited to picking well-defined keywords. It is possible, for instance, to click on the word "observer", and find where that word is first used in the text. This complete cross-referencing makes it possible to find information quickly and to explore related topics without ever resorting to an index or table of contents.

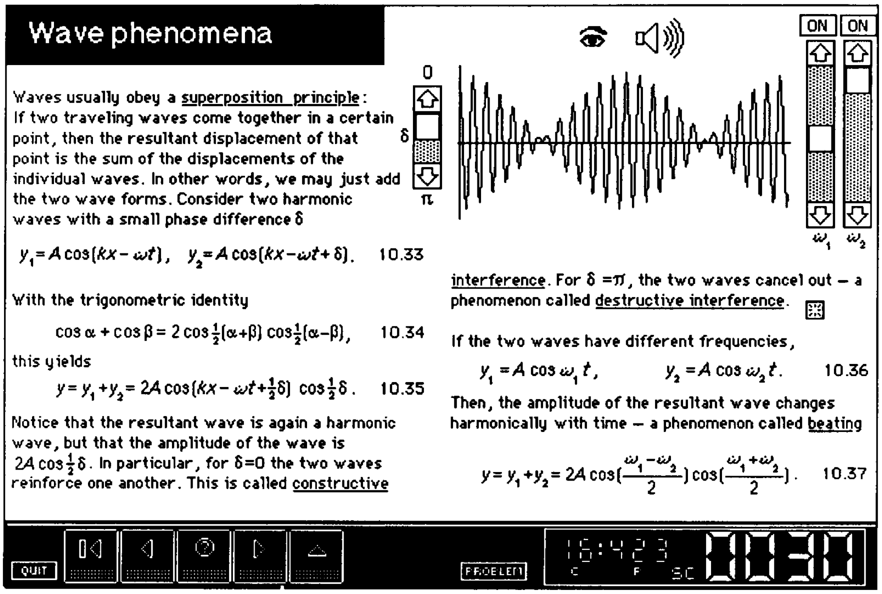

Fig. 3: On this notecard from *Essence of Physics*, the "eye" symbol allows the user to experiment with the superposition of two waves, the frequency and phase difference of which can be adjusted. The program suggests settings that illustrate the phenomena of beats and of constructive and destructive interference. The resulting waveform can also be heard by clicking the speaker button.

The user can also click on symbols in the text or in equations to find the meaning and, if so desired, the definition of the symbol. Similarly, a click on an equation reference immediately brings up the notecard containing the corresponding equation. Many notecards also contain *zoom icons* that, when clicked, bring up windows containing added information on certain subjects. So, the student is encouraged to explore the material by interacting with the software and thus becomes an active participant in the learning process.

The second feature of the program—animations and interactive demonstrations—is designed primarily to help the student visualize and clarify certain concepts introduced in the text. In a textbook, two-dimensional pictures are all that can be printed. However, the computer can add the extra dimension of time or motion, and concepts that may be hard to visualize by conventional means suddenly come to life. Clicking on the eye in Fig. 2, for instance, starts an animation showing the effect of impulsive forces in the collision between two hard spheres. An interactive demonstration treating sound enables the student to experiment with a sound generator and *hear* how waveform, amplitude and frequency affect the tonality, volume and pitch of a sound. Another interactive demonstration (Fig. 3) deals with superposition of waves, and permits the student to adjust the two frequencies ω_1 and ω_2, and phase difference δ, of two harmonic waves, by using the slide bars on the sides of the figure, and to see the resulting waveform. The sound corresponding to the traced waveform can then be *heard* by clicking the speaker button.

Since the students' knowledge of physics in most courses is tested using problems, the program would not have been complete without them. Textbooks offer a combination of example problems with solutions, and problems that the students have to solve themselves, both of which have a number of shortcomings. When the solution is provided, students are constantly tempted to read ahead before they have had time to think about and reflect on the problem. Unsolved problems, on the other hand, do not provide any type of feedback. Even if the answer is provided, the students cannot be completely sure that their own answer is correct until the solution has been reviewed by the instructor. The third and last aspect of the program tries to combine the advantages of these two types of exercises in the form of *interactive* problems.

When I started working on this project, I had only a vague idea of how to implement these interactive problems. I did not want to make extensive use of multiple-choice questions, since they would not adequately reflect the predominant way in which students are examined. Also, I did not want the program to simply verify numerical answers. Firstly, a

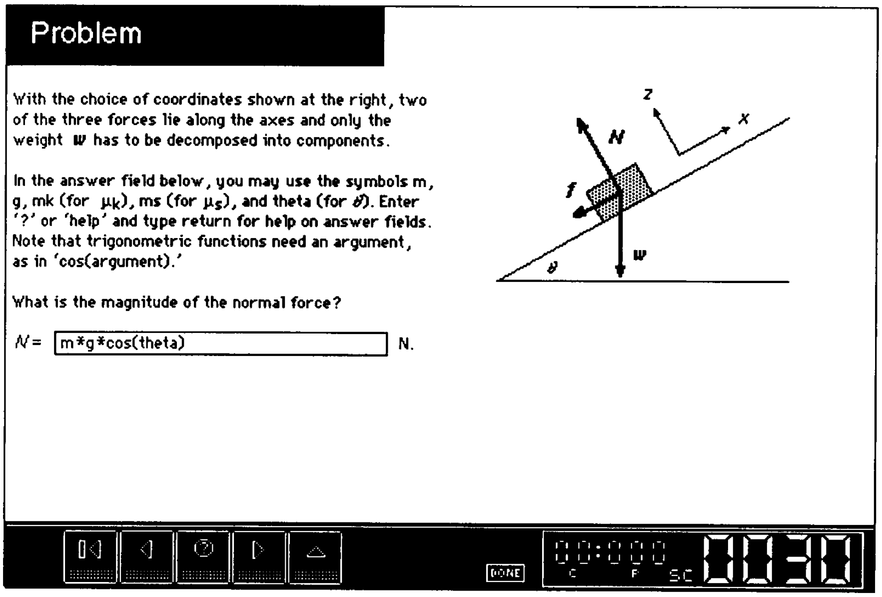

Fig. 4: Section from an interactive problem from *Essence of Physics* illustrating the use of answer fields. Here, the user must enter in the answer field a mathematical expression for the normal force sketched in the diagram at top right. The answer field then evaluates the mathematical expression.

student could get the correct numerical answer even with the wrong derivation. Secondly, and more important, the process of mathematical formulation is an integral part of learning physics. I therefore developed several tools that form the building blocks of the interactive problems.

One tool is the *answer field* into which the students enter their answers (see Fig. 4). What is special about these answer fields is that they can handle symbolic mathematical expressions. In a certain problem, the student is asked to calculate the normal force on a block of mass m on an incline, making an angle θ with the horizontal ($mg \cos\theta$). The student must enter the (mathematically) correct expression into the answer field. The answer field then evaluates this expression, and, in the example given above, would consider any of the following expressions correct: $mg \cos\theta$, $\cos\theta\, mg$, $m^2 g \cos\theta / m$, etc. In addition, it catches some common mistakes and offers specific advice. For example, in the same problem, the answer field would recognize $mg \sin\theta$ as a trigonometric mistake, and tell the student to check the trigonometry. If the student makes a mistake that does not fall into a predetermined category, he/she can try again or get assistance in one of the following ways: receiving a hint, reviewing the material, or seeing the solution.

A second tool used for interactive problem-solving allows the student to use the mouse to draw a force diagram directly onto the screen, as is done in drawing and painting software programs. After drawing each force, the program will ask the student to identify that force. Here, I had to be sure to make the program as tolerant as possible; the gravitational force on a block on an incline can be (correctly) identified by *weight, gravity, gravitational pull* or *gravitational force*. The program will accept any of these and even accept some minor typing mistakes. This is an important feature, because the student might otherwise feel that the answer is rejected unjustly, and therefore become frustrated very rapidly.

A third tool used throughout all problems is a set of routines that keeps track of the student's performance, time taken, type of errors made, etc. When the student finishes a problem, a report card appears which contains specific recommendations (Fig. 5). If, for instance, the student makes a mistake involving friction in a problem on rotational dynamics, this card will suggest that the student review the notecards on friction. Finally, the general performance is compiled on a score card which provides an overview of the student's performance in the various subject areas (Fig. 6).

Many problems are similar to textbook problems. After the problem statement, the student is asked a specific question, which may require entering an expression, picking a multiple-choice answer, or completing a diagram. If the student does not know the answer, the program can either provide a hint or show the solution. Immediately after the first part is finished, the next part of the problem is shown. Most problems contain at least four parts. Several follow a different approach by letting the student make *observations* and take *measurements*. For the topic of sound, for example, the student

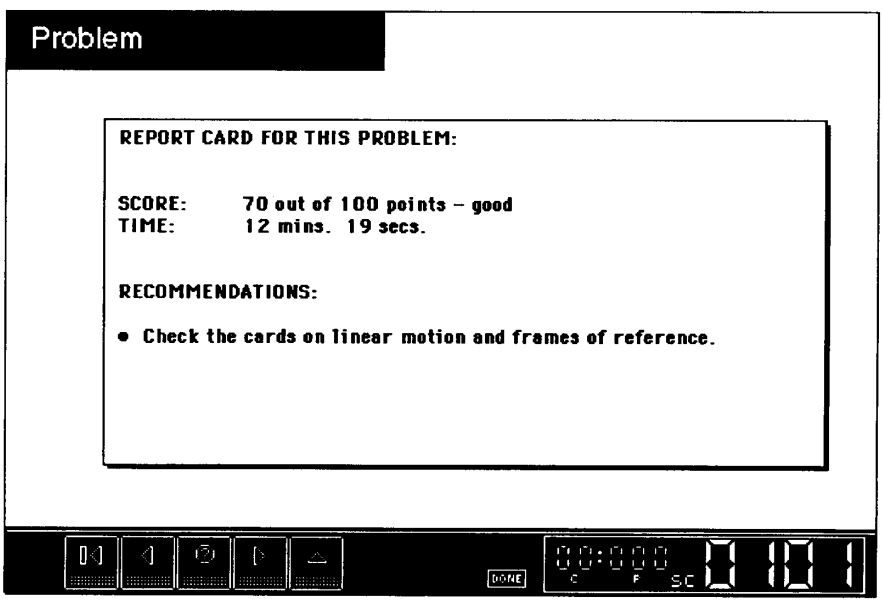

Fig. 5: After completing an interactive problem in *Essence of Physics*, the user sees a report card that lists a number of specific recommendations, based on the user's responses.

presses a button and *hears* a beat. To answer the question stated in the problem, the student must determine the number of beats per second with a digital stopwatch on the screen. In a problem on dc-circuits, the student must use a multimeter to measure voltages and currents in a circuit (Fig. 7). Because of disk space limitations, there are only 24 problems in the current release—one for each topic.

Several colleagues and reviewers have commented on what seems like a large amount of text in the program. Would it not have been better to add more animations or problems? Well, first, I should point out that to add just one extra animation or problem, notecards for several topics would have to be removed, since the notecards take much less disk space. Still, one may well initially wonder about the usefulness of an electronic summary. Contrary to what one may believe, however, this is not the result of a constraint imposed by the publisher—in fact, the editor had frequently asked me to minimize the amount of text in the program. The notecards with text are there *at the request of the students.* While beta-testing the software, I had several students fill out a questionnaire and rate the usefulness of the various aspects of the software. Surprisingly, the tutorial text scored highest, followed closely by the animations and demonstrations. It appears that the students like to have a compact review of the material, combined with the ease of exploring this material at the click of a button.

The entire program design and implementation took place over a period of about one year. I worked on the project, evenings and weekends, over two periods of roughly four months in the course of 1989. I started designing a "shell" by programming the various aspects of the interface. After about two months, I had a functioning shell, complete with notecards for one or two topics, a couple of animations and one interactive problem. It took two more months to enter all the information for the first volume. During the summer of 1989, I tested the software with students from the Harvard Summer School. I then refined the program, and incorporated suggestions I had received from colleagues. The second volume was less work because I used the shell from Volume 1 as a template. Curiously, the most tedious aspect of the project was drawing the illustrations. Using scanned figures in the program was impossible, because of the limited resolution of the screen and the limited amount of available space on the notecards; every figure was carefully composed on the screen, often pixel by pixel, to make best use of the resolution.

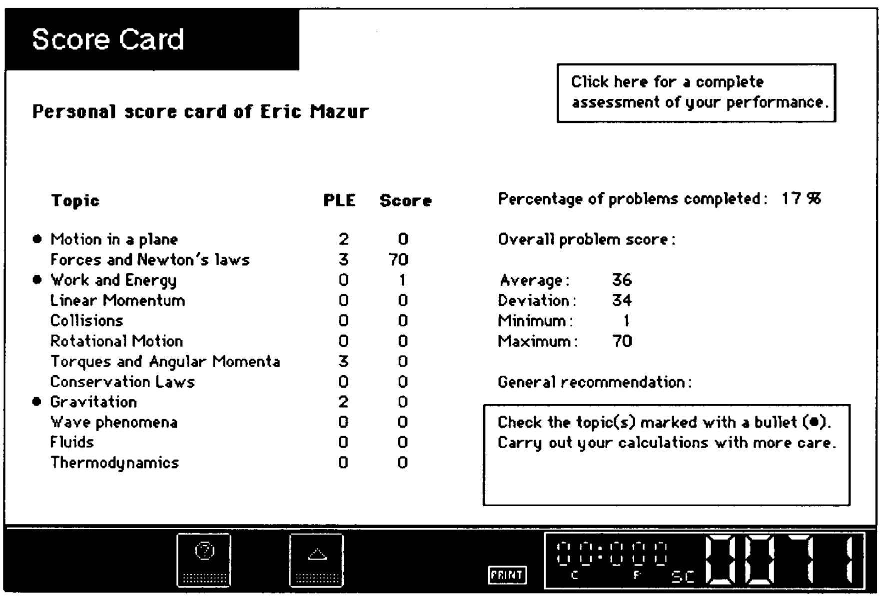

Fig. 6: The user's general performance in the interactive problems in *Essence of Physics* is compiled on this score card. The card shows the user's strong and weak points and provides feedback (not shown here) on a number of specific issue such as mathematical errors, the making of diagrams, etc.

Have I achieved what I set out to do? The students will be the ultimate judge. To a certain extent, however, I was restricted by one of my own design goals—the limiting of the project to two double-sided diskettes, to ensure that the program would run on *all* available Macintosh computers. Because of this, I could not use color or add more animations and interactive problems. There was not enough room on the two disks for quite a few of the animations and interactive demonstrations that I had prepared or planned (deleting notecards would not have helped much since these consist mostly of text and do not take as much disk space). Still, I see this project as a first step to a new *user interface* or *shell* for combining various types of educational software. Many additions and improvements are still necessary. Some related projects I am working on now are described below.

Current and Future Developments

In order to optimize the use of computer technology in education, it is necessary to assess the effectiveness and usefulness of currently existing software as an educational tool. How to carry out such an assessment and how to interpret any results are unclear. After supplying the students in my class with the software during two semesters, I can say that they have reacted positively to the software. For those who enjoy using computers (the majority really), the software appears to stimulate more interest in the subject of physics.

To gain a better understanding of how students are using the software and how to improve the interactive problems, we are now developing a special version of *Essence of Physics* for use in sections taught in a computer classroom as part of the introductory physics classes. The program differs from the released version in the sense that it tracks the students' actions and their answers to problems, and stores this information in a text file. The text files will then be imported into a database for analysis.

We hope to learn two things from the data. The general (action) data will tell us how students use the program, how much time they spend using the various parts of the program, which features they use most,

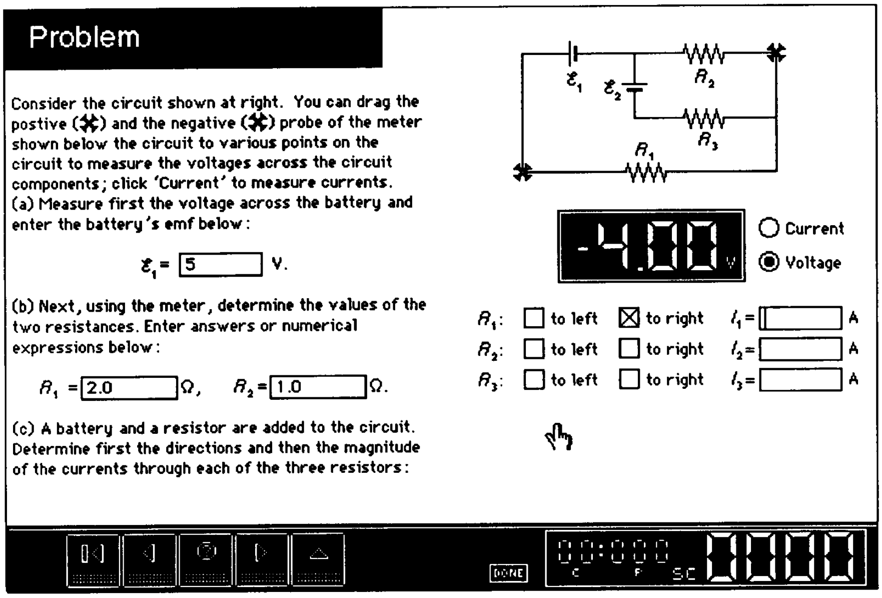

Fig. 7: Computer-based electronics lab. Some of the interactive problems in *Essence of Physics* require the user to "experiment" with computer-representations of instruments. In this problem on dc-circuits, for instance, the user must "measure" voltages across components and currents to answer questions on the circuit.

etc. From the answers to problems, we'll learn what type of difficulties the students have in solving the interactive problems. We will be able to determine if there are any common misconceptions, any misunderstandings because of unclear wording, or any difficulties that arise because of problems with the software, rather than with the actual contents (e.g. because of the way in which expressions must be entered into the answer fields). I consider this information to be very important. It will allow us to improve future versions of interactive problems, because the software can then be programmed to anticipate certain types of common errors or difficulties, and can respond to them in a proper way.

Recent research has shown that students in introductory physics courses have many common misconceptions which conventional instruction is ineffective at correcting.[2] Most of these misconceptions result from common-sense beliefs of how the physical world around us works. Several tests exist for diagnosing misconceptions and determining whether a student will have difficulties with the course material.

This fall, I programmed such a test into the computer and had the students in my class take it as part of the course requirement. The program tracks and classifies the students' responses and provides detailed statistics. The main advantage of a computerized test is that grading and compilation of statistics are completely automatic—the students immediately get to see their results and an analysis of their strong and weak points. As a result, my teaching assistants and I can address several common misconceptions that would otherwise go undetected. On one occasion, for instance, we discovered that although the entire class could recite Newton's third law, a little over a quarter of the students believed that in a head-on collision between a large truck and a small compact car, the truck exerts a larger force on the car than the car on the truck.

To correct these misconceptions effectively, it is necessary to have one-to-one discussions with the students, which require an extraordinary amount of time, even with a small class. We are presently documenting and videotaping these discussions, in the hope of eventually developing a computer program that can not only diagnose, but also effectively correct, various common types of problem in the students' basic understanding of physics.

While not all students entering college today own a computer, it is probably fair to assume that this will be the case before the end of this decade. Also, computer technology and computing power will have increased greatly by then. Since the development of effective course material is much slower than the evolution of computer technology, it is very important to explore newly-available technology even before it is widely available.

Several recent projects, such as the Athena Language Learning Project at MIT, have been undertaken to combine the strengths of computer and video technology in education. The coupling of true video images (still and live) with a hypertext project, such as *Essence of Physics*, offers many exciting new possibilities. Students could, for instance, test their knowledge, pick their own demonstrations, take part in interactive demonstrations, review lectures using a computer, and most important, determine their own pace and sequence while exploring the material.

It will take time to develop all these exciting and potentially useful educational tools. As a first step, I plan to use this new technology in combination with the diagnostic test discussed above. The possibilities are limited only by one's imagination. If a student were to pick the wrong answer due to a misconception, for example, the program could show the student a menu of video clips allowing him/her to see and hear what other students have to say on this subject. Alternatively, the student could view a short lecture or an actual

Table I. Student activities in a classroom setting (left) and in the context of educational software (right).

Classroom	Software
Asking questions	Clicking on words, symbols
Laboratory	Interactive simulation
Sections	Interactive problems

demonstration, which could then be combined with simple questions to keep track of the development of the student's thoughts.

Conclusions

We must not forget that teaching is an art as old as mankind itself. Many techniques from this well-established art can be applied in computer-based education. Let me therefore end by pointing out several parallels between teaching techniques and programming techniques used in educational software.

A teacher must keep the attention of the students, whether in a classroom or in front of a computer screen. The best way to achieve this is to involve students actively in the learning process. In a classroom setting, one may do this by inviting questions. As I mentioned before, I have tried to achieve that in *Essence of Physics* by making it possible for the student to click on words, symbols, references, etc. Clicking takes on the role of asking for more details. One can extend this analogy much further (see Table I). The interactive simulations correspond to the laboratory; interactive problems take on the role of tutorial sections. The advantage of a computer program over conventional teaching is that a computer can supervise students *individually*; in principle, at least, it is feasible to tailor education to the particular needs of a student.

Table II. Parallels between teaching techniques and programming.

Teacher	Programmer
'Passive' lecture or review	Text
Inviting questions	Buttons ('eyes', etc.)
Classroom demonstrations	Animations

These analogies are clearly evident in *Essence of Physics* (see Table II). The text on the notecards corresponds to the standard passive review. The zoom icons, buttons and keywords "invite" the students to further explore the topics at the click of a button, much as a teacher invites questions in the classroom. The animations, like class demonstrations, serve to liven up the presentation and help to clarify or visualize certain concepts.

So, can we teach computers to teach? I believe we are just seeing the beginning of this process and the computer will soon become an integral part of education. Computers will not replace teachers, but they will certainly provide them with an important dynamic tool for improving the quality of education. ■

References

1. E. Mazur, Doing physics with computers, in *Academic Computing*, February 1988.
2. I.A. Halloun and D. Hestenes, The initial knowledge state of college physics students in *American Journal of Physics*, **53** (1985) 1043.