Lawrence M. Bezeau  
Faculty of Education  
University of New Brunswick

Computerizing Programmed Instruction by Parsing Student Answers

Abstract

Programmed instruction involves breaking down and providing information in small chunks called frames, asking frequent questions of the learner, requesting frequent answers, and providing immediate feedback on the correctness of the answers. With the recent widespread diffusion of inexpensive computers and their connection through the Internet, it has become possible to adapt programmed instruction to take advantage of modern personal computers and their communication capabilities.

This paper describes an attempt to adapt programmed instruction to desktop computers using popular web browser software as the presentation medium and the built-in JavaScript programming language as the vehicle for making the programmed instruction interactive. By using the regular expression facility of JavaScript, it becomes possible to parse the answers given by students to ascertain their correctness and to provide helpful feedback to the students in cases where the answers are not correct. The Internet provides a convenient route for students to send a record of their use of programmed instruction back to the author, allowing the author to improve the design of the instruction on a continuous basis. Computerized programmed instruction has proven to be an effective medium for instructing over the Internet.

Programmed instruction was developed by the psychologist B.F. Skinner and became popular as a presentation format for textbooks in the 1960s and 1970s. Since that time there has been a steady stream of programmed instructional materials in book form but this format has not replaced the conventional textbook. Programmed instruction requires that students respond frequently and receive immediate feedback to their responses. For programmed instruction in book form, this feedback is supplied by the correct answers to questions, answers that are hidden from the student until the student responds with his or her own answer. Computers provide the opportunity to take into consideration the actual answer given by the student in providing feedback. Thus both correct and incorrect student answers can be examined by the computer and an appropriate response given to guide the student in the right direction.

The following description is of a prototype of computerized programmed instruction, one example of which can be found at http://www.unb.ca/education/ed3051/c82s01.html. As is the case with any prototype, it is not completely free of programming bugs or other types of errors and mistakes.
**Conventional Programmed Instruction**

A typical programmed textbook consists of a large number of short sequences called frames. A frame normally contains a small amount of information and requires the student to respond in some way, either by answering an explicit question, filling in a blank, or following an instruction. In programmed instruction, answers are normally short, usually not exceeding a few words. After providing an answer, the student sees the author's answer that had been covered by a mask or was on the next overleaf page. Students normally work though the book from beginning to end in a pre-determined order although a few books provide branching that allows students to skip remedial material if they already understand the concept.

Programmed instruction has been controversial from the beginning. Detractors have criticized it for emphasizing memorization over understanding and for valuing behavioural outcomes over cognitive processes. Talyzina (1981) presents a telling critique of behaviourally based programmed instruction arising from empirical research using it to teach concepts of plane geometry to sixth-grade students in Moscow. In her study, as in many others, it is difficult to distinguish limitations in programmed instruction as an instructional technique from deficiencies in the particular instructional program employed to do the research. Carefully written and thoroughly tested programmed instructional textbooks such as *The Analysis of Behavior* (Holland and Skinner, 1961) have proven successful in teaching critical thinking and other higher level skills.

The strong opinions of detractors may have been, in part, a reaction to the exaggerated expectations of the most enthusiastic proponents who anticipated that programmed instruction would ultimately replace most or all other methods of instruction. Langdon summarizes this very well in the introduction to Bullock's volume (1978) on programmed instruction in the multi-volume instructional design library:

> While we have come to our senses in recent years and now realize that PI is not a panacea—no more than any other instructional design—we have also come to realize that its classical (and modern) form has certain valid uses. (p.v).

The instructional design library reflects this philosophy in recognizing many different types of instructional design, none of which is clearly superior to the others in all instructional contexts.

In recent years, programmed instruction has been modified to create specific techniques that allow its use in the treatment of special populations. Errorless learning for persons with memory impairments (Tailby and Haslam, 2003) is an example of such a specific adaptation.
Testing and Quizzing with a Web Browser

There have been a number of recent attempts to automate quizzes and tests in web browsers using the JavaScript programming language (Flanagan, 2002) in pages transmitted over the world wide web. One of the most comprehensive and successful is the “Hot Potatoes” system developed by Half-baked Software (http://www.halfbakedsoftware.com/index.php) at the University of Victoria. Although Hot Potatoes does not implement programmed instruction, it does have modules for multiple choice and short answer questions among others and is fairly easy to use. It also has the advantage of being free for public universities and schools.

Programmed Instruction with a Web Browser

There are several reasons for choosing the web browser as the delivery mechanism for computerized programmed instruction. Virtually all computers have this software, often as part of the operating system, and it allows for simple, fast, and efficient delivery of the content over the Internet. Web browsers incorporate considerable display flexibility and are programmable with the JavaScript language built into all popular web browsers. A fourth advantage is that students can send their responses back to the author over the Internet simply by clicking on a submit button, thus easily providing the author with the feedback needed to improve the page.

This software, once it is delivered by the Internet to the client's computer, operates independently of the web server. This contrasts with the questioning mechanisms in some course management packages such as WebCT (Web Course Tools) in which the answer to each question is sent back in turn to the server for matching against a fixed string. By doing the required processing on the client computer, response time is much faster and the instructional sequence, once on the client's computer, can be completed without an Internet connection. This also allows for the delivery of the software on media such as CD-ROMs. Forwarding the student responses to the page author requires an Internet connection but does not require an email client.

As with conventional programmed instruction, students answer the question or fill in the blank associated with each frame. For computerized programmed instruction this is done by typing a short sequence of words into the computer. With the computer, this sequence of words is ended when the student presses the enter key. Typing an answer into a computer requires the use of the keyboard and makes the use of the mouse for moving from frame to frame an inconvenience. For this reason, frame to frame movement is done automatically once a question is answered correctly or a correct answer is provided by the computer. One completed frame is displayed at the top of the page so that, except for the first frame, the current frame is always the second one on the page. This helps to emphasize the logical sequence of the material and permits the student to easily use
information from the previous frame to answer the question in the current frame. The focus is always on the input box of the current frame so that anything typed by the student is entered into the current frame. The focus is moved at the same time as the page moves up to expose a new frame.

Students are permitted a fixed number of attempts at the correct answer, normally three, after which they are presented with the correct answer. The normal rule of thumb for programmed instruction is that students should get 95 percent of the answers correct on the first try (Talyzina, p. 310). This has proven difficult to achieve, even with conventional programmed instruction, and with computerized instruction percentages tend to be lower because of the presence of typing mistakes and poor spelling which might be considered correct when using a textbook but which the computer normally treats as incorrect. Strategies for dealing with this problem are discussed later. If the proportion of correct answers is too low then the instructional sequence should be revised based on student feedback.

Attempted answers submitted by students are parsed by what are called “regular expressions” (Friedl, 2002), expressions that will match a set of attempted answers and fail to match those not in the set. At least one regular expression is required to match the correct answers and more than one may be used in cases where more than one answer is correct or where the one correct answer is complex. Normally these will be followed by a series of regular expressions to match wrong answers. Associated with each regular expression is a sentence or two of feedback, confirming the correctness of the answer or guiding the student toward the correct answer if the student has provided an incorrect answer. The list containing the feedback is reconstructed each time the student attempts an answer, permitting the feedback to contain this attempted answer. Once a regular expression matches an attempted answer, no more are tried and the feedback for that regular expression will be presented to the student immediately. In practice, most frames will require from two to nine regular expressions.

**Developing Computerized Programmed Instruction**

Unlike a conventional textbook which is simply written, a programmed textbook is expected to undergo a formalized testing process after it is written. Based on its initial presentation to students and their patterns of responses, the frames are revised and new frames may be introduced with the aim of increasing the percentage of correct answers. In the case of computerized programmed instruction, students are encouraged to submit their answers to the author and these are then used to revise the program. This permits the revision process to become continuous; as feedback is received, the author revises the frames.

In writing the sequence of frames the first time, the author attempts to anticipate the correct and incorrect answers that may be received and devises appropriate responses for
these. The actual and potential sets of correct answers will be considerably smaller than the corresponding sets of incorrect answers. Student feedback allows the author to increase the set of anticipated answers and the set of responses for them but the feedback is useful mainly for dealing with incorrect answers since these are more numerous and less likely to be anticipated. There are diminishing returns to this process. As the amount of feedback increases all the common wrong answers will be accounted for. The potential number of uncommon ones is large, difficult to anticipate, and often difficult to respond to appropriately.

For the purpose of writing the software, two types of wrong answers must be considered at the outset, the empty answer and the completely unexpected answer. Empty answers, ones created by pressing the enter key only, are not accepted by this software; students are informed, upon entering an empty answer, that they “must type an answer into the input box.” Empty answers do not count toward the limit on the number of attempted answers. The completely unexpected answer is one that will not be recognized by any of the regular expressions that have been set to parse incoming answers. In any frame, these can be avoided by having, as the last regular expression matching incorrect answers, one that matches any one or more characters. This is desirable when it is possible to give a hint that is specific to that particular frame, in place of a generic response. In this program the generic response begins with the words “Your answer,” followed by the attempted answer itself in quotes, followed by the words “has not been understood.” Obviously, more specific feedback would be much more helpful.

It is not necessary to anticipate all the possible wrong answers but it is highly desirable to account for all possible correct answers in the regular expressions matching correct answers. Failure to do so results in the student being told that a correct answer is incorrect. Recognizing all correct answers means, for example, accepting many synonyms for answers and recognizing that answers consisting of lists can often be equally valid in any order. Feedback from students must be examined carefully for unanticipated correct answers and corrections made to the program as quickly as possible.

The instructor must be alert for answers that are partly correct. For example, if the correct answer is “rights and freedoms” (or “freedoms and rights”) and the student answers with just “freedoms,” the response from the program should acknowledge the partial correctness but ask for both parts of the correct answer in the next attempt. Responding to an answer like “rights or freedoms” requires careful thought. In this case the instructor may wish to simply indicate that the correct answer requires an “and” in place of the “or.”

Another challenge is dealing with spelling and typing mistakes. These are considered together since the computer cannot distinguish between them. After receiving student feedback over several months, it is possible to anticipate most of the common attempted
answers that are probably correct in the mind of the student but that are spelled incorrectly. These can be treated as being correct but to acknowledge their correctness reinforces the mistake and increases the probability that the student will repeat it. Treating it as incorrect may annoy the student and forces the student to answer the question again. My approach, based on experience, is to respond that the answer is probably correct but that it is not spelled correctly. Often I will point out the specific mistake or provide the correct spelling if the word is difficult to spell, but I do require that the correctly spelled answer be entered before the answer is counted as correct. I normally accept incorrect capitalization as correct unless, in the context of the particular frame, the exact capitalization is important. This is handled easily by JavaScript regular expressions which can be specified as either case sensitive or case insensitive.

**Student Reaction**

Student reaction has come through a number of avenues. In my classes, the programmed sequences were optional and were not assigned any mark value. Students were advised to use them if they found them useful as a study aid and to ignore them otherwise. Students have commented informally to me about the system but the only formalized feedback sought so far was the submission of student answers over the Internet. The programmed instruction has been widely used and student comments have generally been favourable. Individual students would often complete the same programmed sequence several times, sometimes to get a higher percentage of the answers correct and, in other cases, at different points in time for review purposes. The most common reaction is that more of the course should be covered in this way.

One specific change made to the software as the result of student feedback was to slow the speed of scrolling from frame to frame. Students initially found that scrolling took place so quickly that one screen was replaced by another in a flash and they were not left with any impression of the relationship between the two screens. Scrolling has now been slowed so that the page can be seen to be moving up the screen and it is obvious that the just-completed frame has been moved to the top of the screen and that the current frame is just below it.

A problem that has arisen with the formalized feedback is that students often do not like to submit an attempt at a programmed instructional sequence if they feel that they have not done well, even though these submissions are anonymous. They may complete a sequence of frames several times and submit only their last attempt. But of course, it is the first attempts that are most useful to the instructor since these contain the greatest number of mistakes. Students have been urged to submit their first attempts and this has occurred with some success.
Conclusions

This research has demonstrated the viability of computerizing programmed instruction and has shown many advantages of the computerized variant over programmed instruction in book form. These include the ability to parse student answers and to respond to both correct and incorrect answers in an appropriate manner. These very advantages place a heavier burden on the author of computerized programmed instruction in comparison to those producing it in book form. Nevertheless, providing programmed instruction over the world wide web permits convenient feedback from the student to the author, allowing the author to more easily improve the instruction.

Bibliography


