SESSION III

THE PLATO IV CAI SYSTEM IN PSYCHOLOGICAL RESEARCH

An introduction to the PLATO CAI system

ROBERT W. HOLT

George Mason University, Fairfax, Virginia 22030

The PLATO interactive computer system is discussed as a fully developed large-scale CAI system. Both the hardware and software aspects of the system are discussed. The discussion of hardware includes the necessary equipment to operate a PLATO system and the flow of control during an interactive session on that system. The discussion of the software on the PLATO system focuses on the TUTOR lesson-authoring language. The aspects of the TUTOR language that facilitate display of material, student input and judging of input, and either student-initiated or author-initiated branching in the program are discussed.

The PLATO system is an interactive computer network designed for computer-based instruction. The PLATO system uses a Control Data Corporation (CDC) computer with extended slow core to handle interactive lessons on up to 1,000 terminals simultaneously. PLATO was originally developed under a National Science Foundation (NSF) grant at the University of Illinois. At present, there are several PLATO systems at universities, as well as several commercial systems run by CDC. Due to this long history of development, there is a large variety of courseware offerings presently available on the PLATO system. Furthermore, the system has been designed to facilitate the additional development of software by the users.

The PLATO system has been specifically designed for computer-based education. The control system and TUTOR authoring language have been designed to facilitate the two major aspects of computer-based education: computer-managed instruction (CMI) and computer-assisted instruction (CAI). Before discussing the useful aspects of this CMI and CAI software, we will briefly review the hardware and operating system characteristics of the PLATO system.

HARDWARE

Hardware for the PLATO system consists of the user's terminal, the site controller, the computer interface unit, extended core storage, and the components of the central Cyber computer. We will briefly discuss each of these in turn.

Terminal

The PLATO terminal is a high-resolution unit consisting of a matrix of 512 by 512 dots that can be sepa-

rately addressed and illuminated. The older terminals used a plasma panel for display purposes and were "dumb" terminals. The newer versions of the PLATO terminal use a cathode-ray tube type of display and are "smart" terminals. The intelligence of the newer terminals can be used to facilitate graphics displays, run abbreviated versions of the TUTOR language, and obtain direct screen prints. Each time a key is pressed on the terminal, the key code is passed on to a site controller for further processing.

Site Controller

The site controller is a multiplexing device that can handle up to 32 terminals at once. The site controller takes the keypress from each terminal, adds a terminal identifier to that keypress, mixes the keypresses from the various terminals, and sends the resulting stream of code to the computer interface unit.

Computer Interface Unit

The computer interface unit combines the input from up to 32 site controllers for input into the central computer. This interface unit adds a site identifier to all the keypress signals from a given site, mixes these keypresses with keypresses from the other sites, and sends the resulting stream of code to the peripheral processing units of the central computer.

Peripheral Processing Units

The peripheral processing units are slave minicomputers that regulate all input to and output from the central memory and central processor in a Cyber computer. The peripheral processing units sort out the keypresses from the computer interface unit and put them in a buffer in the extended core of the computer. Each

terminal has a separate key buffer in this extended core.

Central Processor

The central processing unit polls through the keypress buffers in the extended core. The keypress is actually executed when the keypress and the lesson's code are entered into central memory and executed by the central processor. The advantage of using the extended core as a storage device for keypresses awaiting execution is that swapping from the extended core to the central memory and back again is extremely fast. Although this system may appear to be awkward, it actually is efficient; keypresses are usually executed within .10 sec.

SOFTWARE

The software developed for PLATO consists of a basic student-management system for CMI and a large variety of lessons for CAI.

Computer-Managed Instruction

The software for CMI on the PLATO system allows for testing of students, routing of students to appropriate lessons, and record-keeping routines that keep track of which lessons the student has completed. Other information on the time required to complete each lesson, number of times a certain lesson was reviewed, and so forth, are also available from this CMI software. In general, however, the options available on this system for graphics displays and other features of the PLATO system are not as fully developed as in the TUTOR language used for developing CAI lessons.

Computer-Assisted Instruction

The actual lessons used for CAI or for computer-assisted research on the PLATO system are constructed using the TUTOR language. This language is particularly powerful for displaying textual and graphic material, inputting and judging student responses, and controlled branching through the program. This language is relatively easy to use and can be learned interactively on the system. Help and aids for the programmer are readily available while working on the system. Since the major advantage of PLATO is the fully developed authoring language, we will now discuss this language in more detail.

TUTOR AUTHORIZING LANGUAGE

Display of Material

Standard display functions in the TUTOR language allow the author to display or conditionally display one or more lines of text, arbitrary lines and curves, alternate character sets, and graphs and charts. The conditional display of the material may be based on the value of a program variable or based on interactive student input

from the terminal. Both text and graphics displays can be easily positioned at any point on the screen, sized larger or smaller than the standard size, or rotated to any angle with the use of one command. Erasing parts of figures or blocks of text is also easily accomplished. Animation sequences are typically created by the successive redrawing and erasing of figures on the PLATO screen using these commands in a timed sequence.

Figures of an arbitrary shape may be drawn on PLATO using a combination of lines and arcs or circles. The graphics package that I use allows a cursor-oriented construction of the figure. The graphics package program then creates the appropriate code and inserts this code back in the main program. This method of drawing arbitrary figures greatly reduces the normal amount of time required to program a figure.

Functions may also be plotted on PLATO by specifying the range of the function and the exact expression for the function. These given functional expressions may be of any degree of complexity.

Student Input and Judging

Student input on PLATO is normally accomplished by use of the touch panel or by use of the keyboard. The touch-panel device allows the programmer to specify 1 of the 256 designated squares on the screen of the CRT to represent a given response. When the student touches the screen, a corresponding character is sent to the program, which may then be judged. Typically, the designated areas for the touch panel on the screen are coordinated with a graphics display that informs the student what each area of the screen stands for. The student then touches the area of the screen corresponding to his or her chosen answer.

The keyboard input operates in the standard manner and may consist of single letters or numbers, complete words or sentences, or numerical answers. The options for judging these responses include commands that force judging after a response of a certain length, ignore spelling variations, ignore sequential order of letters or words in a sentence, search for key words that must be present, or ignore optional filler words in the response. For numerical answers, the programmer can specify alternative units for the response that will be considered equivalent in judging the response. For example, the programmer may specify that 100 cm = 1 m, and the student's answer will be correctly judged whether it is phrased in terms of centimeters or in terms of meters. These judging routines may also, of course, be used to judge touch-panel input that corresponds to the words, letters, or numbers displayed on the student's screen.

Branching

Branching in PLATO may be initiated either by the author via the variables in the program or by the student via his or her interactive responses. The author-controlled

branching can use any variable to initiate branching and may branch to any appropriate section of the program. For example, the results of a proficiency test may be used either to proceed to new material or to direct the student to review old material.

Student-initiated branching may use any one of several function keys on the PLATO terminal (help, lab, data, next), as well as the normal keys. These branching commands must be inserted by the programmer, but once inserted, they take effect only if the student makes the required keypress. The student-initiated branching is particularly useful in providing students with optional review of old material, further explanations or clarifications of old material, or additional new material to cover. The use of both program-

initiated and student-initiated branching in the PLATO system allows for a very flexible approach to controlling the student's progress through the lesson.

SUMMARY

The PLATO system is a very well developed computer-based education system that is also quite useful as a research tool in applied settings such as psychological research. Many of the aspects of this system that have been designed to facilitate authoring of CAI lessons in general also facilitate the presentation of instructions and response alternatives in a research setting. The following papers will address the use of PLATO in several types of applied settings.