XS-Ø: A SELF-EXPLANATORY SCHOOL COMPUTER

J. Nievergelt
H. P. Frei
H. Burkhart
C. Jacobi
B. Pattner
H. Sugaya
B. Weibel
J. Weydert

Institute for Informatics
ETH
Zurich, Switzerland

Abstract

XS-Ø is a low-cost interactive system that serves as a self-explanatory school computer. Particular attention has been devoted to making the man-machine dialog easy to follow for the inexperienced user. The system includes a course on computer programming, a programming system for writing, editing, executing, and debugging programs interactively, and a filing system containing private and public libraries. The language offered to the user is a version of Pascal. The system is realized on a small stand-alone computer which supports a small number of graphic terminals. This hardware consists of the most cost effective components currently on the market.

System Overview

In the fall of 1975, a project was started at the ETH Zurich with the aim of developing an interactive system that should serve as a self-explanatory school computer with the following properties:

- A user should be able to learn programming without further help.
- The system serves as a tool to write, edit, execute, and debug programs.
- A filing system provides private libraries for users and for groups of users (e.g., classes) that need to share files.
- The software of the system enables the development of course material on subjects other than computer science.
- The system runs on a dedicated computer and is able to provide instantaneous response to most of the requests that might occur.
- The system is portable in the sense that one person may easily move it from one place to another.
- The hardware consists of the most cost effective components currently on the market.

Particular attention has been devoted to making the dialog between the user and the system easy to follow for the novice. Thus a program called EXPLORE guides the user through the space of all services offered, and provides information on all options available to the user at each site. The space of sites contains a public library of lessons and procedures, and a register of all users of the system. The register gives access to the private libraries of users, and serves as a means of communication among the community of users. The public library contains a collection of lessons and exercises of programming, and an inquiry system which, when given a keyword, provides a short explanation and refers to lessons that cover this concept.

An extended version of PASCAL-S is used as both an author language for the preparation of the course material and a programming language for teaching purposes. This language is slightly larger than PASCAL-S, with an expanded set of standard procedures (for string manipulation, graphics input-output and timing).

The presently used hardware consists of a PDP-11 V03 with 28K words of memory, a dual floppy disk drive, and two graphics terminals, each consisting of a TV monitor, a microprocessor Intel 8080, and 8K bytes of RAM. The system software is written in MODULA, a higher level system programming language. Only the parts of the system that are running in the terminal (graphics routines and parts of the editor), are written in assembly language.

The following pages show snapshots of the screen as the user is engaged in a variety of activities.

References

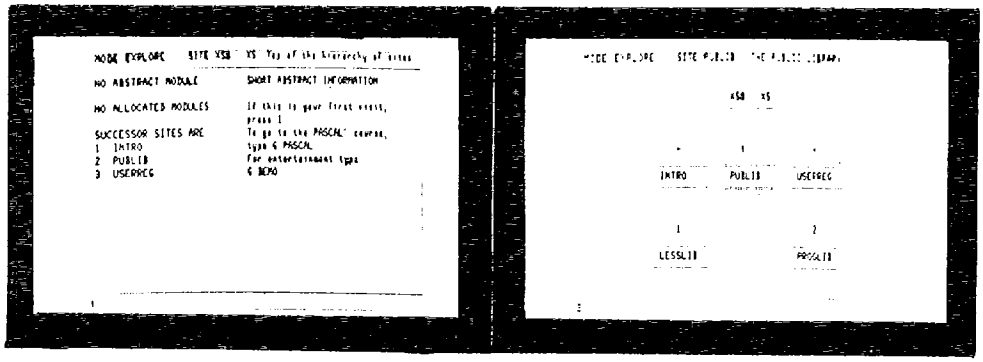
The design of XS-Ø has been heavily influenced by the Automated Computer Science Education System [1, 2] developed on the PLATO System [3] at the University of Illinois.

1. J. Nievergelt et al. "ACSES, An Automated Computer Science Education System," Angewandte Informatik 4/75, pp. 135-142, April 1975.

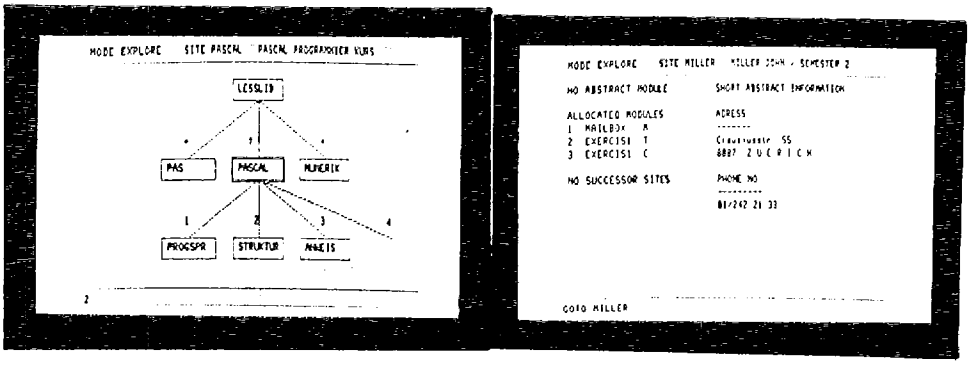
2. J. Nievergelt. "Interactive Systems for Education--The New Look of CAI," Proc. IFIP World Conference on Computers in Education, pp. 465-471, North Holland Publishing Co., Amsterdam, 1975.

3. D. L. Bitzer. "The Wide World of Computer-Based Education," Advances in Computers, Vol. 15, pp. 239-283, Academic Press, New York, 1976.

Exploring the Space of Resources

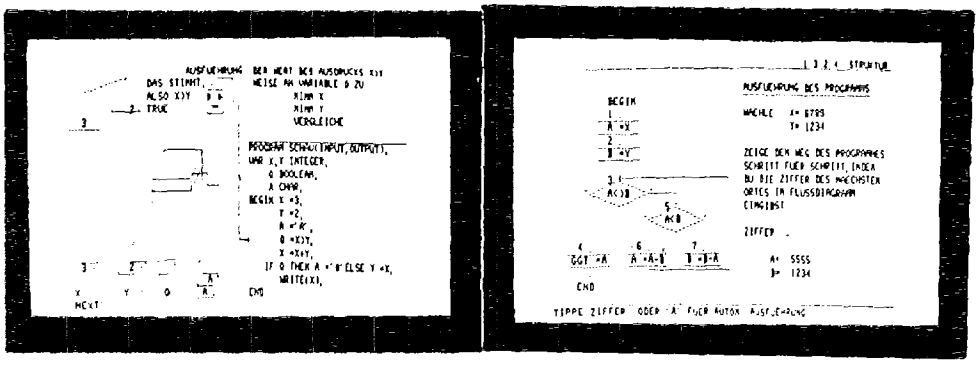


The resources of XS-Ø are organized in the form of a tree. The user enters the system at the root (left figure), where the novice has a choice to see an introduction to the system and its use. The user can ask for a pictorial representation of the neighborhood of the current site he is at (the public library in the figure at right).

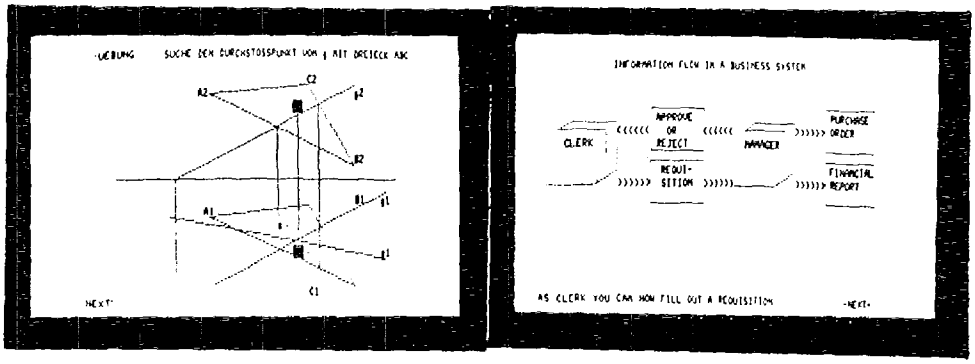


With a single key press, the user moves to the father, left brother, right brother or any son nodes of his current site. Since the screen changes instantaneously, he can explore the space of sites rapidly. In the figure at left, he has found the Pascal course in the public library. Every user has his home site in a region of the tree called the user register (right). There his private files are attached. Groups of users (e.g., classes) also have home sites, where shared files may be attached.

Sample Lessons

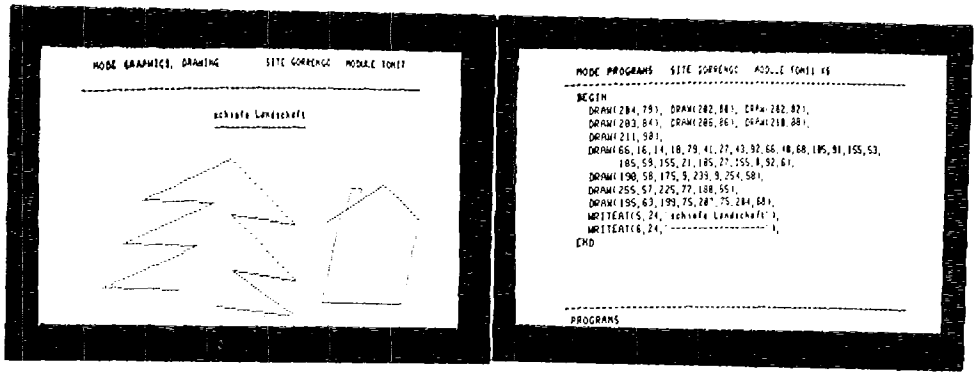


Two snapshots from the programming course. At left the program is executed line-by-line. The value of an affected variable travels between storage (boxes at lower left) and the computer's processing units (the balance, for example, is a comparator). At right the student has to indicate the location of control as the flow chart for computing the greatest common divisor of two integers is executed.

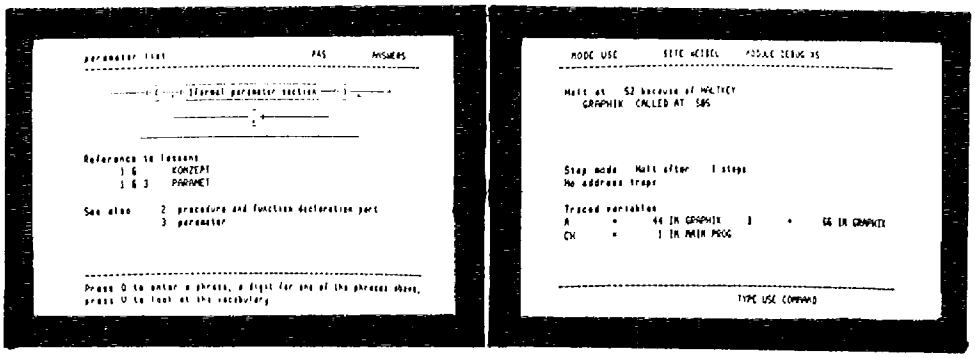


At left the student constructs the intersection of a line with a place in a lesson on descriptive geometry. He moves a cursor to enter points. In the business simulation at right, the student first plays the role of a clerk who fills out a requisition form, then of the manager who approves or rejects each item requested; he watches the flow of forms that travel through an office system.

Program Preparation and Execution



with the graphics editor, a user draws pictures and enters text by moving a cursor on the screen (left). The program editor automatically generates Pascal statements (right) that draw this picture when executed. With a text editor, the user includes these statements in his program. The program editor can also generate an entire Pascal program that steps through individual frames produced by the graphics editor, thus serving as a simple CAI author language.



An on-line manual for Pascal displays syntax diagrams (in the figure at left for "parameter list") and refers to lessons in the public library that explain the concepts involved. Figure at right: A user can interrupt a program in execution at any time by pressing the HALT key, then asking for the current value of any variable. If he specified that certain variables should be traced, their value is displayed automatically.