"WISE" AUTHORING OF SOFTWARE
by M. Lynn James*

Historically, there have been several impediments to the widespread adoption of computer-based teaching in chemistry. The high cost of hardware, the fact that few chemists have the necessary expertise to develop and/or adequately utilize software in chemical education, and the availability of relatively little quality software have been among the principal impediments. With the advent of the microcomputer and the resulting drop in price, the cost of hardware has been considerably reduced. Efforts by individuals and groups, such as the workshops sponsored by the Committee on Computers in Chemical Education and Project SERAPHIM, have educated many chemists in the use of computers. Some very good software is currently available but this has required enormous effort by a few very dedicated programmers. Time estimates for producing such software can range from 50 to more than 300 hours of programming for each hour of useable software. Obviously, not every qualified programmer is willing to put in such time.

In addition to the expertise and time commitment necessary, a major hinderance to the future production of a large amount of quality software is the tedious nature of programming in BASIC, Pascal, assembly and machine languages typically used for developing much of the software. The task is particularly tedious when graphics are included in the program. One approach in alleviating this problem is the use of an authoring language. Authoring languages come in a variety of forms ranging from ones, such as PHOENIX, that are designed for use on main frame systems to those, such as Super Pilot, which work on micros.

Typically, those designed for use on main frames (1) do not give access to the full capability of the computer, (2) tend to optimize some important subset of machine capabilities, (3) are based on higher level languages with powerful commands available for generating text and graphics displays, accepting and judging responses, and carrying out a variety of branching, logging and score keeping functions, and (4) can become difficult to use as enhancements and additions are made to the basic syntax to increase their power. Furthermore, being main-frame based they are not useable in a classroom or laboratory setting.

On the other hand, authoring systems implemented on micros can be easily taken into the classroom or laboratory and are typically menu driven and therefore easily learned and used, but they are not too powerful in terms of the number and sophistication of the display and logical functions that they support.

A compromise that has the best features of the main frame and microcomputer authoring languages without many of their disadvantages is accomplished through hybrid authoring languages such as WISE (WICAT Interactive System for Education). This specific authoring language is menu driven, highly prompted, easy to learn, highly productive, uses a sophisticated instructional strategy which involves complex displays and logic that can be implemented much more quickly than in other authoring languages, and gives a range of options and ready access to a wide variety of programming utilities and tools that offer the user complete access to anything the computer can do.

Characteristics of WISE are its power which allows the author to perform any function of which the computer system is capable as part of any screen display (frame) in any lesson. It runs on a 32/16 bit processor supported by a true multi-user, multi-tasking operating system. WISE allows for three levels of authoring. The first level uses a highly prompted set of menus and requires no coding. This level allows for the presentation of information and free-response or multiple-choice types of questions, each allowing for the incorporation of a wide variety of graphical displays.

The second level of programming, which is based upon a Pascal-like language, allows for the types of calculations important in chemistry and includes the above features but in addition permits conditional branching, use of thousands of system variables (string, real, integer, long integer, and pool), calling the logic and display of other frames as subroutines, control of external devices such as videodisc players, and manipulation of common variable pools accessible to any set of simultaneous users.

The third level involves an External Program Frame Mode and enables the author to use anything the computer hardware and software is capable of doing as part of any frame of any lesson.

WISE includes a powerful Character Set Display Editor which allows for the following features:

1. The author is able to define an unlimited number of special symbols or character sets, consisting of any number of key assignments, down to the single pixel definition level. All such symbols can be varied in size and are moveable and rotatable by the Display Editor with no requirement for coding.

2. Sophisticated displays can be generated and manipulated including graphics, text, and animations.

3. Primitives include dots, lines, boxes, circles, center-point arcs, three-point arcs, text (in any font size, or orientation), and fitted-spline curves.

4. Modifiable attributes of any graphics objects include line width, line pattern, and different types of fills.

5. Manipulation of the order in which objects appear on the screen can be easily accomplished.

6. Any group of objects can be defined as a "segment" and treated as a single object without losing the capability of manipulation of its separate parts.

7. All objects can be scaled including scaling to negative values for mirror-image effects.

8. All objects can be copied into any other frame of any lesson, rotated, moved, and animated including multiple object animation with scaling, motion along defined paths (a very attractive feature for use in chemistry), and rotation, leaving a trail or not, saved in author defined graphics libraries and bit mapped.

9. Digitized audio, videodisc and videotapes displays may be included.

Function Key Editor features allow the author to define any set of keyboard and numeric keypad keys as special function keys with a great variety of purposes.

Student Response Modes may be typed in as alphanumeric responses, or as screen locations through predefined keys or touch screen, and time limit features for responses can be easily included.

Answering Processing includes the ability to search for key words ignoring all or certain other words, unlimited synonym dictionaries may be defined, and spelling tolerances may be set by the author.

Through these features, productivity of quality software is greatly increased and authoring requires much less skill in programming and becomes a much more rewarding experience.

*Department of Chemistry
University of Northern Colorado
Greeley, CO 80639

ELECTROLYTE DEMONSTRATION
by John Hnatow*

The purpose of this demonstration is to replace or to supplement the conventional light bulb conductivity apparatus used in general chemistry demonstrations. If one has access to microcomputers in a chemistry setting, this demonstration will process the collected data in a very efficient and organized way and allow students to conceptualize a very important theory. Most chemistry instructors not familiar with simple interfacing techniques will be quite surprised at both the simplicity of the programming and the low cost of the materials. I developed this program at the Allentown College/Moravian Academy Microcomputer Interfacing Curriculum Development Institute in 1984 and have had time this school year to refine it.