

APPLICATIONS OF TECHNOLOGY IN TEACHING CHEMISTRY  
An On-Line Computer Conference  
June 14 TO August 20, 1993

PAPER 15

MENU DRIVEN PROGRAMMING FOR STUDENTS AND TEACHERS

AUTHOR: Reed A. Howald  
Department of Chemistry and Biochemistry  
Montana State University  
Bozeman, MT 59717  
(406)-994-5415  
uchrh%planet.dnet@terra.oscs.montana.edu

SCHEDULE: Short questions on this paper: July 30, 1993  
Discussion of this paper: August 12 and 13, 1993

ABSTRACT: Menu Driven Programming for Students and Teachers

How does one control a computer for the collection of experimental data in an instructional chemistry laboratory? Computers are useful for the collection of data, for data analysis, for presenting material, and for examinations precisely because they are versatile. However this means that they can do nothing for us without detailed instructions given faultlessly. A few incoming students know some programming, but most students and chemistry teachers do not. One way for our students and colleagues to tell their computers what it is that they want the computers to do is through "control files".

Control files are ordinarily in unreadable binary, but they can be constructed from numbers and text, with sufficient English to be interpreted by human beings. One can write a menu system which even beginning students can use to put together such control files without syntax errors. Two generations of such systems will be demonstrated.

TEXT OF THE PAPER:

We cannot expect to teach even one standard programming language (BASIC, FORTRAN, PASCAL, or C++) along with teaching chemistry, and our students and colleagues need a way to direct their computers to do what they decide they want. Some students come to college with a thorough understanding of BASIC programming honed through hours of work on Apple IIe computers. But there are still large numbers of students in freshman chemistry with little or no computer literacy. In any case it is somewhat of a shock to students to come to a chemistry laboratory and find that they are expected to use a computer to collect and analyze chemical data.

At Montana State University we introduced computers and interfaces into our freshman chemistry laboratories in 1987. We hoped that this would speed up and simplify data collection and analysis. We expected that the versatility of the system would allow a wider range of experiments at a lower overall equipment cost. These goals were certainly realized, along with an unexpected benefit: the students had time to think about the meaning of their data as it was being collected

and analyzed.

We hoped that having computers in the laboratories would simplify and speed up the process of providing them with good computer assisted instruction on the topics in freshman chemistry lecture which were particularly difficult for them. Progress in this area has been much slower than I personally anticipated, but I hope to show in this paper that menu driven programming can be used in the development of good computer assisted instruction.

At first I expected that we would need a separate computer program for each experiment scheduled. I actually wrote some of these and they were used in the laboratories in Autumn quarter 1987. However it was clear then that a better system could and should be developed, with one executable program and separate control files for the various experiments. This concept was developed with input from many individuals. I certainly made some contributions, and substantial credit goes to Dr. John Amend, Bruce Ivey, Kathleen Tucker, Glenn Howald, Dr. Eric Hood, Allen Overcast, and Andy Cohen.

The menu oriented program for making the control files (BUILDER.EXE) was not ready in December 1987 when I had to finish the laboratory manual for winter quarter 1988. I believe I am the only individual who has made and tested a substantial number of control files by hand. It is a difficult job.

However with BUILDER.EXE available we found that TA's, freshman students, and even high school science teachers can create useable control files. We are indebted to Dr. Ron Furstenau who first showed that freshman chemistry students could be taught this system in a one quarter laboratory, and then assigned individual projects taking up to three laboratory periods to complete.

Figure 1 shows a student in the MSU freshman laboratory in the process of changing a control program. Demol illustrates the individual steps in this process. It is reasonably complex, but each step involves using arrow keys to select a menu item, and then pressing the "enter" key.

Figure 1 was taken with a video still digital camera in our freshman chemistry laboratory. It is available as a 320x200x256 color mode 19 picture in GIF format and in a 640x192 monochrome (mode 6) version. Other pictures showing the computers and interfaces in use can be made available. The menus used by this system are all in text mode, but they all use special IBM symbols and are not easy to capture, display, or print. The two most important menus are saved as demp0014.txt and demp0011.txt and can be examined even if the software to run demol does not run correctly on your computers.

This system has been in use for six years at Montana State University. It has changed over time, but is recognizably the same from 1988 through 1993.

The software has been licensed to SCI Technologies, Inc., 1716 W. Main, Bozeman, MT 59715. It is currently available from them in conjunction with hardware purchases.

At this point a question arose as to whether customers should have access to the source code if they needed it for a particular application. Eventually the group answering "no" received licensing rights to the current software. Those who answered "yes, access to the source code can be purchased" faced the problem of redeveloping the software.

This led to a careful look at the shortcomings and potentialities of the system shown in demo1. Some possible problems were listed. Fifteen choices was not enough for some applications. The use of arrow keys can be quite slow. All menus should be of the same type. Menus ought to be as easy to revise as control files, i.e. they do not belong in the source code.

On the positive side an effective menu system could do much more than build and use control files. It could be a way of presenting information and testing knowledge; it could serve as an authoring system for computer assisted instruction.

When machine graded multiple choice exams were first developed the machinery had limits which made it difficult to deal with more than 5 choices. In this situation educators decided that 5 was the optimum number of answers to give for each question. Unfortunately this limited number encourages students to learn to guess wisely instead of learning the material thoroughly.

My personal experience with multiple choice exams strongly favors supplying 10 to 15 answers for each question, and offering partial credit for answers partly incorrect. Thus the new menuing system was designed to offer up to 15 selections in one menu, and using a single letter was not sufficient for a selection. Using two letters, with 26 squared or 676 options was chosen as a better system to implement for fast choices from a menu. This new software is far enough along to show its potential. It is illustrated in demo2, and both demos were written with the new system. The menus are normally presented in graphic mode 6, and two typical ones are in Figures 2 and 3.

To run either demo you will need an IBM compatible computer with EGA, VGA, or better graphics. Follow the directions given for this conference for receiving encoded text files. Use UUDECODE to convert runrn.exe and saved screens (\*.sc\*, \*.px\*, and \*.txa files) back to binary code. The other files sent, 8x8.fnt, calibrat.dat, \*.mn\*, \*.rn\*, and \*.tx\* are used as text files and can be examined directly, however using UUDECODE to remove header information is recommended. With all the files copied to a 1.4 Meg disk or to a subdirectory on your hard disk, you can view the demos by typing RUNRN DEMO1.RN <return> and RUNRN DEMO2.RN<return> respectively.

If you are interested in seeing the new system in operation without using UUDECODE, the files will be supplied on floppy disks. Please request them by e-mail to Dr. Howald, giving a mail address and specifying the type of floppy disk you want (5.25 or 3.5 inch and low or high density). It is very easy to edit the \*.tx\* files to change the information placed on the screen, and the numbers in demo2.rn can be adjusted to change the size and position in which the text of demo2.tx appears on the screen. The font file is limited to 8x8 pixels, but individual characters can be edited and changed in any way

you want. I hope that the ability to print anywhere in any size on IBM compatible screens will have many classroom uses. The important question at this point while the system is still being developed is: Exactly what would you like to be able to do with it? I am hoping for a lot of widely ranging suggestions from this computer conference.

Reed Howald