MESSAGE FROM THE CHAIRMAN:

The last ten years in education have seen a great deal of emphasis on acquiring and implementing the microcomputer into the K-16 curricula. Despite this emphasis, extensive and meaningful uses of the computer in education is still quite spotty and limited. What the next ten years will bring is still in question. In an article, "Infusing Computing Into the Curriculum: Challenges For the Next Decade", which appeared in the April, 1989 Academic Computing issue, David L. Smallen, Hamilton College deals with this question. In the material which follows, I have tried to summarize his more salient points.

Although there are many reasons that might be given to justify a more extensive infusion of computers, Smallen gives what he considers to be three fundamentally sound reasons: 1) the emergence of computer science as a true liberal art, 2) the need to prepare students for life in the information society, and 3) the potential that computing has for improving the instructional process. With respect to the second point, Harlan Cleveland, Dean of the Hubert Humphrey School of Public Affairs at the University of Minnesota stated "by the end of the century, approximately two-thirds of all work will be information work ... (computers) empower those who learn to use them to make complex judgments in the more mindful knowledge of alternative futures ... People who do not educate themselves-and keep reeducating themselves-to participate in the new knowledge environment will be the peasants of the information society." For the third point, the computer offers the possibility of individualizing instruction and simulating the real world, and, in doing so, the learning process can be made more interesting, challenging, and effective.

Smallen suggests seven challenges that stand in the way of revitalizing and recreating educational disciplines through the infusion of computers into the curriculum.

I. **Provide appropriate recognition and incentives for faculty to improve the process of instruction.** The lack of adequate reward structures for faculty software developers is a major road block. We see a parallel example to this problem in the traditional lack of college or university support for helping faculty become better teachers in comparison with support for research. Solutions to this lack of recognition lie in such things as having the process of evaluating software similar to the way other professional activities are evaluated and in providing national recognition for quality software and courseware that is developed.

II. **Access, pricing, and distribution mechanisms for instructional software must be improved.** Faculty must have convenient access to information about such software that includes at least a list of names of others that have used the software in a similar setting and how to contact them. Reasonable pricing is essential so that students can afford to buy it, when asked to, and so that colleges can afford to license sufficient copies for their legitimate use. Distribution mechanisms must be set up to make it possible for faculty to obtain "desk copies" of software similar to how textbooks are handled.

III. **Develop more effective planning processes for instructional use of information technology resources.** Informal planning methodologies dealing with instructional technology typically used in the past must be replaced with methods that maximize the impact of it in relationship to the individual institution's mission. The time has come for us to do more than talk about planning, we must make a commitment to do it.

IV. **Encourage the formation of consortia of universities and colleges to deal with instructional applications of information technology.** One of the great strengths of higher education in the U.S. is the diversity of institutions. Consortia need to be built on the various strengths of different institutions, such as coupling the incentives of the teaching institution with the technical expertise of the research university to develop teams of educators working on the problem of developing high quality software. Consortia can consider some of the significant hurdles present at smaller institutions including lack of technical expertise, access to information channels, and access to more favorable pricing for software.

V. **Emphasize the role of the computer as a general learning tool.** Students need to learn more software tools to help them throughout their undergraduate experience and beyond. This includes word processing, general purpose tools for problem solving, database managers, tools to improve their ability to read critically, an opportunity for creative thinking, and others.

VI. **Ethical and legal use of software must become the norm.** Institutions of higher education must make a commitment to the legal and ethical use of software, thus creating an atmosphere on campus that will instill in their students a respect for the law and a desire to act in an ethical manner. In part, this is accomplished by institutions licensing sufficient copies of software for expected usage levels thus eliminating the incentive to copy software illegally. In turn, vendors must recognize these efforts by making instructional software affordable. Further, those responsible for computer services should adopt policies of not helping people with software problems if it is clear that the software is being used illegally.

VII. **Information technology services organizations must create supportive environments.** Smallen suggests that this should include standardizing hardware and software configurations on campus to eliminate wasted time dealing with technical interfacing problems, setting up effective software libraries or network servers for public facilities to reduce the cost of instructional software implementations, setting up classrooms equipped with large screen projectors and hookups to tie into the computer systems used in instruction, and working out standards that can be used in planning new public computing facilities.

With the varied uses of computer technology that can be made in chemistry we, as chemical educators, can play an important role in this infusion. In the words of John Kemeny, a recognized pioneer in the field of using computing for instruction, "Once you succeed in integrating the computer into the classroom, you will find that your entire style of teaching changes, and I can assure you that you will never go back to the old fashioned method..."

Editor

Choosing a new computer. After 3

years of considering different computer systems and working on extra projects to get the money to buy a new system, I finally bought a new one. It wasn't easy or obvious which to choose.

My first need was for a good system to write programs on since that is my primary interest. Having transferred programs successfully between many systems I knew it could be done, but usually with much time and effort. Since I'd rather spend this effort on programming, another requirement was to develop programs on a system available to my students. So we had to have the system available on campus. It also had to be easy for very poorly prepared freshmen to use. It would be helpful to have many excellent programs available. Needed were the best in programming languages and tools, great graphics (I think the weaker the student the more important the graphics are). I needed a fast system since I don't have much patience waiting for computers to work. I'd like any new neat innovations in hardware or software to be available for my machine. It would have to be expandable.

My first thought was a Sun. I worked with one for quite awhile learning C and UNIX. Quite impressive in most departments, but too complicated for my freshmen. Also we don't have any available for students.

The Apple II systems don't seem powerful enough, but extra hardware keeps coming out and we have many around campus. Many students also own their own. But they are pretty slow and the programming tools are not the best.

IBM and especially clones were appealing. You can get enormous amounts of great software, add on parts, and if a clone would serve your purposes a hefty discount from IBM prices. Almost any new idea in software or hardware would have to be available eventually because of the large number of IBM PC's around.

Macintosh also looked good because large amounts of great software, add ons, etc. But no Mac clones so the prices looked quite formidable.

At this point I was thinking about an IBM 70 or 80, and Mac II,. but leaning pretty heavily toward an IBM clone because of the price. Just to be sure I called some colleagues at Princeton to see what they thought. One physcist somewhat, to my surprise, was heavily in favor of the Mac. His responsibility is purchasing systems for a micro lab used by students and faculty and he travels to all the computer shows. Many vendors love to say Princeton is using their stuff so he gets many previews of programs and hardware long before the rest of us mortals.

He said the Mac interface is so consistent across applications that once you have learned the Mac interface you have about 80% of each new program. In his opinion Mac software is 3-4 years ahead of similar IBM software. This probably isn't true for programs written by the same company for both machines, but it does seem to take more than windows and a mouse to make a good useable interface.

Further research with colleagues in chemical industry brought similar comments. Even graphs showing significant decreases in time required to learn new programs on the Mac. Time to learn new applications on IBM systems supposedly doesn't decrease as much because each is quite different from the others. This was a big point for me since I've seen my family struggle getting work done on my old system when they had been off for a while. I just couldn't imagine my average student putting in the time and effort to learn a new set of commands for each program. In fact since I've been using computers with my students for over 12 years I know they won't learn new programs without a lot of effort on my part.

So I started looking seriously at the Mac II. Prices still bothered me, but I found that list prices have little to do with what you pay. Discounts range from 25-50% from list! Since prices are changing constantly and the discounts you can get depend very much on your personal situation, I won't even try to quote prices. First decide on the equipment then shop around for the best prices. It really makes a difference. Your best discounts are usually available thru your college, state purchasing plans, a relative who works for a company that gets sizeable discounts.

I decided on a Mac IIx with a 80 meg hard disk, the new high density diskette drive (1.4 meg) which can also read and write to IBM formatted disks! I also got A/UX the UNIX like operating system which is new to the Mac. Working with it for about 6 months now, I'm getting pretty comfortable. So far it has pretty much lived up to its billing. It does take awhile to learn and longer to program, but my learning time on each new well designed (follows the Mac guidelines) program is shorter and shorter. I find it very intuitive and friendly. You can experiment instead of reading manuals and you can usually do what you want without crashing the system.

My experience with my students has been excellent. I asked my students how many had experience with the Mac - it was less than 5%. I arranged to bring them into the Mac lab and showed them how to start the Mac and bring up and use a program I wrote. Within 5-10 minutes pairs of students at each Mac were successfully doing and learning about problems. I was of course very pleased. If new users can be up and running with this little, effort the future is looking more promising. Comments from students ranged from 'it's a lot of fun' to 'it was almost too easy to learn' (these are students talking about chemistry!!!). Not a single negative comment. Yet.

My own experiences are similar. My programming tools are the most advanced available. They are much more complicated than the average wordprocessor or spreadsheet. The Mac interface is so intuitive that I can lay off for 6-8 weeks and start right back as if I worked yesterday. My family also loves it.

Problem: I have A/UX 1.01 (Apple Unix) up on my machine. It takes 65 meg of my hard disk and I cannot run a single program I have under it. This leaves under 15 meg for my stuff on a 80 meg hard disk. Although I'm probably developing some good habits in throwing old stuff away I'd really like more of my disk storage available. Version 1.1 A/UX showed up recently, on something over 30 diskettes. I looked at the rather large amounts of documentation and decided I may need a lot of time to set it up. UNIX is powerful but it is difficult to work with. If I can get any info on the new version I may try it, but I'm seriously considering erasing A/UX and starting over without it. I've not found any good sources on using A/UX , and this makes the lack of understandable documentation a major problem. Most Mac ,documentation is very good but not for A/UX.

Recommendations: 1) Chemists generally need complicated word processing so the enhanced keyboard is very worthwhile.

2) Once I filled a 650 meg disk in 3 weeks so I feel a big disk for storage is useful.

3) I got used to a large screen monitor while using the Sun, so I have a 20 inch Moniterm Viking 2/72. At the moment I have no pressing need for color. The large size screen is much more useful for almost everything. I'd get a no glare coating for the screen. Very helpful. I use a standard IBM screen for this newsletter and it probably takes 8-10 times longer because of the constraints of the screen. All you need is to win a lottery to pay for the equipment.

NOTE:

Chemtext is a chemist's wordprocessor from Molecular Design (415-895-1313). The third figure down on the cover was done with ChemText. I find it difficult to use, nonintuitive, crashes, loses material, and has some of the worst documentation I have seen. I suppose if you used it a lot you'd get used to it. Personally I can do everything I need in far more pleasant ways. If anyone has used it extensively and likes it feel free to send a positive review.

```
SOURCE FILE: FLAM1
------ NEXT OBJECT FILE NAME IS FLAME.OBJ0
7C00:             1          ORG  $7C00
0000:             2          OBJ  $7C00
7C00:             3 *
7C00:             4 ********************
7C00:             5 * FLAME
7C00:             6 ********************
7C00:             7 *
0025:             8 VTAB     EQU  $25
0024:             9 HTAB     EQU  $24
6200:            10 SET1     EQU  $6200     ;REGULAR CHARACTER SET
6200:            11 SET2     EQU  $6200     ;ALTERNATL CHARACTER SET
7C00:            12 FLIKLOC  EQU  $7C00     ;VTAB AND HTAB OF FLAME LOCATION IN
7C00:            13 *                            THESE TWO BYTES
7C02:            14 FLAG     EQU  $7C02     ;STOP FLAG
7C03:            15 TOTCNT   EQU  $7C03     ;A COUNTER - WHEN TOTCNT IS ZERO
7C00:            16 *                       THE ROUTINE EXITS
7C05:            17 BURNCT   EQU  $7C05     ;FLAME PICTURE TO BE DISPLAYED
7C00:            18 *
7C00:            19 *
7C00:            20          DS   16        ;SAVE 16 BYTES FOR CONSTANTS
7C10:            21 *
7C10:A9 0B       22          LDA  #11       ;VTAB OF FLAME LOCATION
7C12:8D 00 7C    23          STA  FLIKLOC
7C15:A9 09       24          LDA  #9        ;HTAB OF FLAME LOCATION
7C17:8D 01 7C    25          STA  FLIKLOC+1
7C1A:A9 00       26          LDA  #$00      ;ZERO THE STOP FLAG
7C1C:8D 02 7C    27          STA  FLAG
7C1F:8D 03 7C    28          STA  TOTCNT
7C22:A9 E2       29          LDA  #$E2      ;THE SECOND FLAME PICTURE
7C24:8D 05 7C    30          STA  BURNCT
7C27:A5 25       31          LDA  VTAB      ;SAVE LOCATION FROM WHICH
7C29:48          32          PHA            ;THE ROUTINE WAS CALLED
7C2A:A5 24       33          LDA  HTAB
7C2C:48          34          PHA
7C2D:            35 *
7C2D:            36 *
7C2D:20 83 7C    37 ROUND    JSR  LOCFL
7C30:            38 *
7C30:20 06 62    39 FLIKER   JSR  SET2
7C33:AD 05 7C    40          LDA  BURNCT    ;LOAD CURRENT REPRESENTATION OF FLAME
7C36:C9 EA       41          CMP  #$EA      ;THIS IS THE LAST FLAME PICTURE
7C38:D0 05       42          BNE  NO
7C3A:A9 E1       43          LDA  #$E1      ; LOAD FIRST FLAME PICTURE
7C3C:8D 05 7C    44          STA  BURNCT
7C3F:20 ED FD    45 NO       JSR  $FDED     ;PRINT FLAME PICTURE
7C42:20 6C 7C    46          JSR  DELAY
7C45:EE 05 7C    47          INC  BURNCT    ;GOTO NEXT FLAME PICTURE
7C48:EE 03 7C    48          INC  TOTCNT
7C4B:D0 12       49          BNE  ROO
7C4D:            50 *
7C4D:20 83 7C    51 EXIT     JSR  LOCFL
7C50:A9 A0       52          LDA  #$A0      ;BLANK FLAME
7C52:20 ED FD    53          JSR  $FDED
7C55:68          54          PLA            ; RETURN TO CALLING LOCATION
7C56:85 24       55          STA  HTAB
7C58:68          56          PLA
7C59:85 25       57          STA  VTAB
7C5B:20 00 62    58          JSR  SET1      ;REVERT TO NORMAL CHARACTER SET
7C5E:60          59          RTS
7C5F:            60 *
7C5F:20 72 7C    61 ROO      JSR  KYBD      ;CHECK KEYBOARD
7C62:AD 02 7C    62          LDA  FLAG
7C65:C9 10       63          CMP  #$10      ;IS FLAG SET?
7C67:F0 E4       64          BEQ  EXIT
7C69:4C 2D 7C    65          JMP  ROUND
7C6C:            66 *
7C6C:            67 *
7C6C:A9 20       68 DELAY    LDA  #$20      ;STIRRER AND DROPS WILL GO HERE
7C6E:20 A8 FC    69          JSR  $FCA8
7C71:60          70          RTS
7C72:            71 *
7C72:AD 00 C0    72 KYBD     LDA  $C000     ;WAS A KEY PRESSED
7C75:2C 10 C0    73          BIT  $C010     ;CLEAR KEYBOARD STROBE
7C78:C9 D3       74          CMP  #$D3      ;LOOK FOR S KEY
7C7A:F0 01       75          BEQ  STOP
7C7C:60          76          RTS
7C7D:            77 *
7C7D:A9 10       78 STOP     LDA  #$10      ;SET FLAG IF S WAS PRESSED
7C7F:8D 02 7C    79          STA  FLAG
7C82:60          80          RTS
7C83:            81 *
7C83:AD 00 7C    82 LOCFL    LDA  FLIKLOC   ;SET TABS TO FLAME LOCATION
7C86:85 25       83          STA  VTAB
7C88:AD 01 7C    84          LDA  FLIKLOC+1
7C8B:85 24       85          STA  HTAB
7C8D:80          86          RTS
7C8E:            87 *

*** SUCCESSFUL ASSEMBLY: NO ERRORS
```

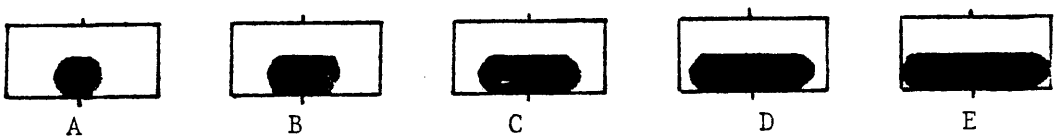

Figure 1

An Animation of Distillation
Part II
The Stirrer

This is the second of a series of articles which describe how to program an animated graphic for distillation. The complete graphic shows a flickering flame, a rotating stirrer bar, falling liquid, and a slowly increasing liquid distillate. In Part I the production of the flame was described. In this part, we explore the illusion of a spinning bar. It is assumed that you have a copy of CHEMUTIL-2(1) available for your Apple II series microcomputer.