



## FROM ZERO TO SIXTY IN SEVEN MONTHS: MANAGING THE CHANGE TO A NEW COMPUTER SYSTEM

Terry A. Ward  
Academic Computing Services  
University of Northern Iowa  
Cedar Falls, IA 50614

A NEW SYSTEM ?! The very words strike fear into even the most hardened of user consultants. Such is the nature of system implementation and conversion that even grown analysts have been known to prefer the job of IBM JCL debugger to that of managing the change to a new (and heaven forbid), another manufacturer's mainframe.

This presentation will outline some of the steps taken at the University of Northern Iowa to make the transition from one computer system (HP-2000) to another (Harris 800). Before I begin, I must clear up some common misconceptions: we are the University of Northern Iowa. Mail addressed to Ohio or Idaho is not likely to reach us; and yes, there are three state universities in Iowa!

By way of background, the University of Northern Iowa has a student body of approximately 10,000 and is primarily noted for its emphasis upon teacher education. (This relates to its early history as the "State Teacher College of Iowa"). However, as we will soon see, the computer users today come from all colleges and disciplines.

The computer situation at UNI (that's us) has evolved over the years beginning with a simple remote job entry facility tied into the University of Iowa's mainframe. A few years ago, the University acquired a Hewlett-Packard HP-2000 system that allowed for interactive BASIC processing. The link with the University of Iowa was maintained, and in fact is still in use today. As time progressed, it became apparent that more computer power was required in-house, and so the decision was made to acquire a Harris 800 super-mini system that would allow both interactive and batch processing for the UNI user community.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Association for Computing Machinery. To copy otherwise, or to republish, requires a fee and/or specific permission.

© 1982 ACM 0-89791-088-5/82/011/0115 \$00.75

With this decision made, how does one go about planning, managing, and recovering from the inevitable misteps of such an implementation. This paper will outline our methods, successes, and most interestingly, our failures.

The first task, of course, is defining the problem. The situation at our university is somewhat unique in several respects. Firstly, the Harris computer would be used solely for academic computing. The administrative needs were and are being handled by IBM systems in a separate facility. This obviously helps the situation somewhat. There is not the pressure for conversion that there might be on a mixed administrative/academic facility. The problem of payroll and student records conversions immediately come to mind as headaches that are thus avoided. Our luck could not last however, and problems were soon to surface.

As I mentioned previously, the University has a very diverse range of computer users that must be accomodated in any system conversion. On the one end of the spectrum, we have the computer science and mathematics faculty that simply want access to the new system. On the other end, we have such "technically naive" users as the editor of the North American Review as well as those people who have used the current systems for CAI and who are unconcerned with the "CPU's heritage."

A second, somewhat related issue that must be faced in this conversion was the quantum leap from the HP-2000 to the Harris 800. Those of us on the technical side of data processing tend to underestimate the conceptual problems encountered in this type of change. To us, "a computer is a computer is a computer" and we simply substitute new syntax for old. However, the situation from the user perspective is not so simple.

Imagine yourself in the situation of a fairly naive user who has grown accustomed to the HP-2000 (or comparable machine at your installation). Every single command issued is a BASIC command. His accessing of the operating system to list a disk directory is actually handled as a BASIC command in the immediate execution mode. Everything one does is done in the only environment available on the HP-2000. That environment is the BASIC one. There is never any chance that he will enter any other environment where commands will be misunderstood.

Now imagine, if you will, this same (now luckless) individual presented with a computer like the Harris 800 (the image of an IBM mainframe will certainly make these points even more vivid). Here, our poor user is presented with a plethora of computing environments. Before even accessing the system, he must make decisions dependent upon his terminal type and computer mainframe desired to inform the port selector of his wishes. Upon entering the system he needs to know the appropriate Job Control to even access his once familiar BASIC. Should one be unfortunate enough to type RUN MTH1 from the Job Control environment, they will be presented with the following "meaningful (?)\" comments:

ER 361

or, with the extended error message option:

ER 361: JOB CONTROL ERROR: PROCESSOR NOT FOUND

To one accustomed to the rather pleasant and charming facilities of a single language system (such as BASIC on the HP-2000), the switch to this multi-lingual contraption is likely to be painful without suitable help.

Finally, there comes the last major problem at this macro-level to be considered. Namely, the problem of the University Computing Services staff itself. Here, we are faced with the dilemma of having to become the "experts" immediately and this problem is only accentuated by this particular choice of computer mainframe -Harris (?!). At the same time our user base is expanding rapidly, we are faced with a totally new system for the staff as well.

I mentioned the growth in user demands. A few examples might suffice to highlight this point. In 1982, UNI graduated its first computer science major. This semester, there are over 300 people enrolled in the introductory Pascal class (on the new system, of course). From 1979 to 1982 we have witnessed the growth in time-sharing capabilities go from 51 terminals (fully two-thirds DECwriters) to over 200 (now less than 15% hardcopy). The number of courses that expect to use the new system this next semester is approximately 90 and these classes represent approximately 4500 student users. Lastly, in this area of grisly examples comes the estimate of computer usage made in 1979 for this year that missed the mark by approximately 350% in terms of estimated CPU hours and connect time.

Into this maze of rapid change and increasing user demand, to paraphrase a popular saying, "What's an UCC (or Univ. Computing Center) to do?"

## **The Solutions:**

### **Documentation Solutions**

Obviously, the first area that must be attacked is that of documentation (another dreaded word in the computer field). As can be seen from Figure 1, one step in this process of educating the user community involves the establishing of what we call documentation levels. Basically, we define the intended audience for a document and then see what type of documents are necessary for each type of user. At the Reference end of the spectrum, we are simply providing materials of a sufficiently technical nature that experienced personnel can function. These are either locally produced or reproduced documents comparable to the system documentation supplied by Harris.

The second level of documentation produced is that aimed at the less experienced user and aimed at basic tasks. Sample documents from this level

include a tutorial in the use of text processing capabilities. Because of the nature of the environment (i.e. JCL, text commands and the like), this document contains material from several different system environments. However, the unifying theme of these documents is that one interested solely in text processing (e.g. the editor of the North American Review or the secretary of the Religion and Philosophy Department) can gain valuable instruction through such a document. Similar documents exist for statistical processing (the typical audience here would be a graduate student doing research for a thesis).

Thirdly, we have available documents that highlight specific system areas and are done in a very compact form. For example, we have pocket guides for MINITAB, for the Job Control or for SPSS on the Harris.

Finally, we have available the "Quick Notes" that are limited in scope to one or two pages and deal with such specialized topics as the preparation of job files to access the SAS statistical system at the University of Iowa via our remote entry system.

In addition to these computer-related publications, there are an essential number of procedural documents available. As an example, UNI processes approximately one quarter of a million test sheets each year in its test scoring service. An entire document exists (and is distributed to all new faculty) detailing the facilities and capabilities of this service. Such a document is useful not only for documentation purposes but also for a nice introduction to the computer services facility in general.

In addition to all this "old-fashioned" material a certain amount of information is available on-line for rapid access. While not exhaustive, we do have documentation on-line for late news items and known software bugs. Also, the text processing software has on-line help facilities.

## **Software Solutions**

A second major area of solutions lies in the area of software. One obvious help in this respect is the relative simplicity inherent in the Harris Job Control (vis-a-vis an industry giant's version). COPY means copy; LIST means list; the examples could continue, but this has made our task somewhat easier.

The greatest help has however been in the capability provided in the Harris system for the construction of what it calls "macros" or files of Job Control instructions that can be made to operate interactively. This macro capability is somewhat similar in scope and power to the SSG, "Symbolic Stream Generator" software available on larger mainframes.

Two examples of the usefulness of such a capability for machine conversion will be presented.

In the first, we have the problem of making available a fairly extensive library

of CAI material (from numerous suppliers) to an entire range of users varying in their sophistication. As can be seen from the sample run presented in Figure II, the user interacts at a fairly high level with the machine and need never concern himself with the Job Control limitations (such as program names, file assignments, etc.) Macros at this user level exist for such things as the entire public library documentation system, the CAI library, or the text-printing portion of the entire text-processing application.

At a somewhat different level, we can use macros to make certain types of user tasks simpler. Let us take the task of sorting a data file for example. A user will often want to sort a particular set of data and in many cases has no knowledge of FORTRAN or the way to call and use the sort utilities. Because the Harris macros can be made to operate interactively, we can construct a system as shown in Figure III, which simply guides the user (rather painlessly) through a file sort.

Obviously, with all these great intentions and publications, we still have the task of teaching users the system. In a sense, you can lead the users to the documentation but you cannot make them compute!

### Training Solutions

Our initial effort at user training was somewhat like this presentation -a hopefully entertaining and not too boring presentation on material. In a manner uncannily like this talk, little was accomplished. A few people attended and learned some things, but the basic task of "computer evangelism" was still a hope and not a reality.

Our second effort, begun in mid-semester and continuing into the semester interim, was much more successful and one I would recommend highly to any academic computing facility.

In this second try at user training, we decided to hold training sessions on Saturday mornings. Approximately one month to six weeks in advance of these "hands-on" training sessions, we circulated brochures describing the courses and their prerequisites to all faculty members. To allow for maximum attendance these sessions were scheduled for Saturday mornings. At this time of the day, the computer was relatively free from student use and terminal clusters across campus could be utilized. Our initial concern over attendance (Saturday morning from 9 to noon?) proved fruitless.

To date approximately one-quarter of the entire instructional and professional staff have attended at least one of our training sessions. The attendees have ranged all the way from graduate students to an academic dean (certainly no small feat in itself!)

There are, of course, certain incentives for attendance. Firstly, accounts are established to the attendees that are valid for one month and that are without charge. While most people use this for training purposes, a certain percentage

use this "free time" for research. As far as we are concerned, once they are hooked on the system, we don't really care if they learn by doing our exercises or by doing their own research. A second incentive provided in these Saturday sessions is that they are somewhat structured. Before we begin our terminal work we present a lecture of basic ideas and, if appropriate, hand out materials for use (e.g. pocket guides, assignments to complete, etc.)

This process of "hands-on, cost-free" introductions to computing capabilities is going to be continued this next semester, albeit with an added twist. All faculty members have been given \$100 of computing time and it is hoped that this might entice some otherwise reluctant users to try our services. One area that is particularly likely to be used is text processing. As an aside, one of our largest group of users is in the Religion and Philosophy Department, all of whom were introduced to interactive text processing at one of our "hands-on" sessions.

Finally in this area of solutions, I might mention one that has just recently been implemented. When the Harris was initially installed, the software manuals were available from the University Book & Supply Store at costs ranging from five to thirty dollars. Few manuals were sold at these prices and in addition, there developed intolerable delays in procuring manuals from Harris. A typical class might want twenty BASIC manuals and there would be a one-half semester delay. To allay this situation, we obtained permission to reproduce the documentation locally. Manuals are now available at prices ranging from approximately three and one half to ten dollars. Since we are now in the publishing business, the problem with delays is gone. A helpful hint to those of you who might like to persuade your mainframe manufacturer to approve such a suggestion. After presenting the wholly logical and (to our mind) completely compelling reasons why we should be in the publishing business, we just happened to mention similar situations were Harris' competitors (such as DEC) had made similar arrangements. I'm not sure which half of our argument clinched the case, but permission was soon granted and we now publish user documentation locally.

## **The Failures**

Obviously any task such as converting to a new system is bound to have its share of failures, and this process at UNI was no exception. There are basically two areas that I feel are open to improvement, not only at UNI, but at any given university computer facility in the country.

The first concerns documentation of programs available for public consumption. The rate at which utilities are written and installed greatly exceeds our capability to present documentation to the user. Though alleviated somewhat by on-line facilities for documentation, the problem remains.

The final problem area is that of "late-breaking" news. Our newsletter is published quarterly (due to personnel limitations), and there is no rapid means of communication of available software updates.

## The Future

Just when things are settling down and the top of your desk reappears, the specifications change. At UNI, our future projects include the installation of a new port-selector for the system (and thus new terminal user documentation, new training sessions....). Also, a new graphics laboratory is being established and I can already hear the requests for documentation and programs.

Oh, about the title -- we are now up to sixty CPU hours of usage per month, not bad for a NEW SYSTEM.

## FIGURE I.

### (Documentation Schema)

- I. Reference Level (20 - 40 pp.)
  - A) Job Control
  - B) Text Processing
  - C) Statistical Processing
  - D) Test Scoring
- II. Tutorial (5 - 20 pp.)
  - A) Job Control
  - B) Text Processing
  - C) Statistical Processing
  - D) Terminal Guide
  - E) Procedural Guide (procedures not software)
- III. Handy Reference (1 - 4 pp.) [ACS specific info]
  - A) Job Control
  - B) Statistical Packages (SAS, BMDP, SPSS, MINITAB)
  - C) Remote Job Entry
  - D) Language Processors
    - 1) APL                      2) BASIC
    - 3) COBOL                4) FORTRAN
    - 5) Pascal
- IV. Manuals
  - A) Harris Language Manuals (approx. 100-200 pp.)
    - 1) APL                      2) BASIC
    - 3) COBOL                4) FORTRAN
    - 5) Pascal
  - B) Harris System Manuals (approx. 100 - 200 pp.)
    - 1) Job Control            2) Utilities (SORT, TEXT, FORMAT)
- V. On - Line Documentation
  - A) News Files
    - 1) H800 News            2) HUE News
    - 3) STATNEWS            4) PLIVNEWS
    - 5) GRIPES
  - B) Documentation
    - 1) CONDUIT              2) MINITAB
    - 3) TEXT



FIGURE II.

(Educational Program System - CONDUIT)

[An example of a very simple macro application]

>CONDUIT

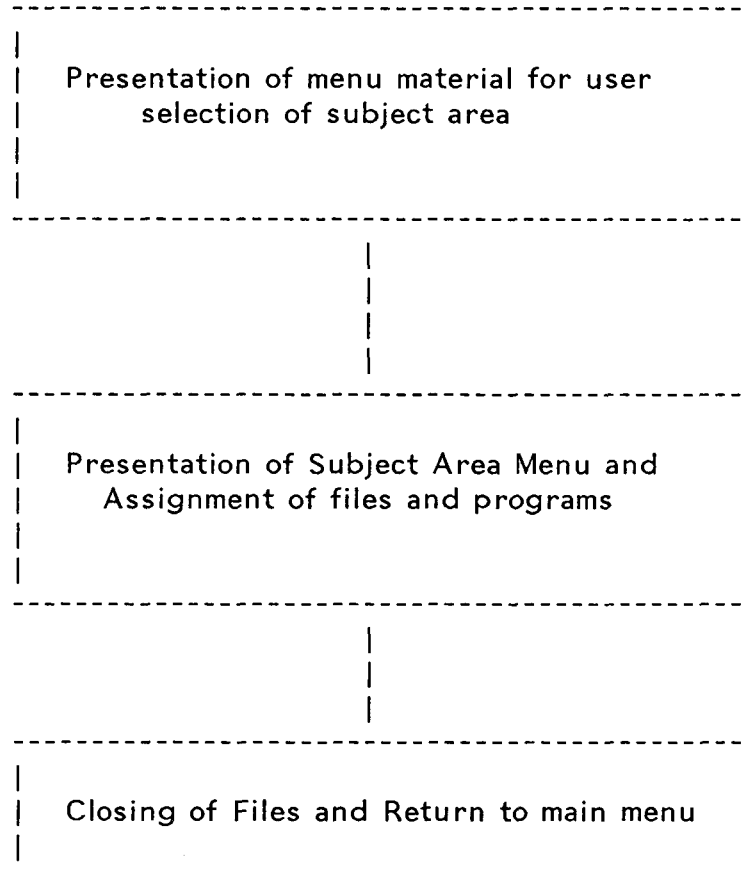


FIGURE III.

(Public Sort Utility \*P-SORT)

[An example of a complex application simplified with macros]

>\*P-SORT

