JULY/SEPTEMBER 1983

VOL. 6, NO. 3

JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED
JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

CONTENTS

PUBLICATIONS COMMITTEE MEMBERS

Publisher
William M. Crow

Art Production:
John Bird, Anne West

Typesetting
LARC Computing Inc.

HP 3000 International Users Group
289 S. San Antonio Road, Suite 205
Los Altos, California 94022 USA
415/941-9960

JOURNAL
OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

HP 3000 INTERNATIONAL USERS GROUP

BOARD OF DIRECTORS

Preface

Lloyd D Davis
Associate Editor, HP3000 IUG Journal
and
Chairman, Special Interest Group for Education (S.I.G.E.D.)

The Publications Committee at the International Meeting of the HP3000 IUG in Montreal in May, 1983 requested that two Special Interest Groups (SIGs), those for Education and Medicine, should each contribute an entire issue of the Journal from their respective fields. Accordingly, this issue is composed, with one exception, of articles representing activities associated with the HP3000 and its use in educational environments.

The use of HP3000 in educational sites is varied as users include both academic and administrative applications; many installations serve both types of users. Sites include school districts representing grades kindergarten through 12, vocational training institutes, junior colleges, colleges, universities, consortia, research institutes, and other institutions with substantial training missions.

A request was mailed to all 650 readers of the quarterly SIGED Newsletter, encouraging them to contribute articles for this issue of the Journal. About a dozen articles resulted from this mass appeal and from other individual requests. When viewed from the very tight timetable that was necessary, the authors did not have more than several weeks in which to submit articles. Hence, I believe those authors submitting should all receive a vote of thanks from the readers of this issue of the Journal.

This issue has papers in the HP3000 administrative areas of education, representing topics such as inventory control, library circulations, materials on curricular planning and test creation, KSAM as an alternative to IMAGE, a popular student records system in use in California, and a management information system from Iowa. Academic topics include the use of the HP3000 in dietary analysis, a popular career guidance system, the use of a tree data structure for supplying computer services information at Bryn Mawr College in Pennsylvania, use of HP3000 in the management programs at Purdue University, the analysis of literary text, and the simulation of CP/M and micro assemblers on the HP3000. An article on Hewlett-Packard's efforts to support computer literacy at the high school level in California is included.

Some educational sites are experiencing difficulty in finding one single machine that can handle all of their tasks; a Southern institution has presented both a philosophy with which to view that problem and policies to manage such a multivendor computer center.

The papers come from a variety of sources. Geographically, the states of Ohio, Tennessee, Pennsylvania, Oregon, Kentucky, Texas, Indiana, Iowa, Louisiana, and California are represented. Several kindergarten through grade 12 school districts, four or five small colleges, several regional universities, a major Mid West university, and a large West Coast junior college system have contributed papers.

Recently, HP's management made a public commitment of support for present users and active and aggressive marketing efforts for new and expanded use of the HP3000 within education. With over 350 educational users in the United States, Canada, and Mexico plus another 100 or so others scattered over Central and South America, Europe, Africa, Australia, and Asia, HP has built a broad, sound base upon which to build even more sales in education. The scope and breadth of the papers in this issue give some sense of what is happening within the HP3000 educational community that SIGED represents and serves. Other SIGED members and I hope that you are as excited and pleased about this Journal Issue as we are. We invite all to submit articles on their activities to either the Journal or to the appropriate SIG newsletter. If we can accomplish this threefold goal of authoring of materials, creating an awareness of educational usage of HP3000s, and fostering SIG participation, we will have been successful.

JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

Automating Academic Library Functions: A Case Study

Randolph L. Whitson
Assistant Director of Libraries
University of Tennessee at Chattanooga Library
Chattannooga, Tennessee 37402
Phone (615)755-4469

Background

Early in 1980, the University of Tennessee at Chattanooga (UTC) Library appointed an ad hoc committee of librarians to investigate the possibility of automating the Library's circulation functions. This project was initiated as part of a five year grant to the UTC Library from a prominent local foundation. At that time the UTC Library was serving a student population of 8,000 along with 2,000 other users. The Library's collections totalled some 800,000 volumes represented by 200,000 machine-readable records. The Library's annual circulation was averaging 65,000 transactions at one service location on the UTC campus. Both the Library administration and the Library staff felt that the manual circulation system which was in operation at that time was terribly deficient and overly cumbersome to patrons.

The library committee already knew that computers had been used for a number of years and in many different libraries to control the circulation of library materials. Moreover, recent reports in the literature seemed to indicate that the trend toward automating this and other library functions was increasing and that a wide variety of minicomputer systems was becoming available from an ever-increasing number of vendors. Although their review of the literature initially revealed that automated systems were not only expensive (both to initiate and to maintain), the committee did find several good reasons to automate a library's functions and to automate them as soon as possible. These included:

1) Cost containment (especially if a library like UTC's was growing rapidly and if computer costs continued to drop and labor costs continued to rise);
2) Dramatic service improvement (especially in terms of inventory control, file maintenance, notice and bill preparation, statistics gathering, management analysis and report generation);
3) Showcase for the library and the parent institution (especially in terms of introducing students, faculty and staff to computer technology in a practical operating environment).

The committee also found that libraries which had automated certain functions often incurred savings which were used to improve other public services or to introduce entirely new services. Both of these options were of particular interest to the UTC Library.

Following their literature review, the committee drew up the project outline listed below and immediately began work on a set of detailed specifications for an automated circulation system which would operate more efficiently and effectively than the existing manual system and one which would easily interface with other library functions. The Library's ultimate goal was eventually to build a totally integrated library system which would automate all of the Library's functions.

Project Outline

Phase 1 / Preliminary planning

Tasks
- Conduct a needs analysis
- Prepare justification for automation
- Survey the market
- Begin preparation of library staff

Costs
- Upper & middle management staff time
- Clerical & support staff time
- Travel for on-site visits
- Purchase of reference materials
- Duplication, postage, telephone, etc.
- Preliminary training
- Consultant fees (optional)

Phase 2 / File conversion

Tasks
- Survey options for bibliographic file conversion
- Justify, choose and implement best choice
- Plan patron file conversion

Costs
- Upper & middle management staff time
- Clerical & support staff time
- Equipment
- Archival tape manipulation, loading and testing
- Site modification
- Forms and supplies
- Barcode/OCR labels
- Consultant fees (optional)

Phase 3 / Choosing a system

Tasks Prepare specifications
Evaluate proposals
Negotiate contract

Costs Upper & middle management staff time
Legal assistance
Duplication, postage, telephone, etc.

Consultant fees (optional)

Phase 4a / Implementation: Management

Tasks Appoint project manager
Supervision of project by project manager

Costs Project manager time
Clerical & support staff time
Training
Duplication, postage, telephone, etc.

Phase 4b / Implementation: Site preparation

Tasks Prepare computer room
Prepare terminal sites

Costs Technical consultant fees (optional)
Air-conditioning and humidity control
Utilities
Electrical connections, cables, conduits, etc.
Furniture
Noise and static control
Security for equipment
Telecommunication equipment (if necessary)

Phase 4c / Implementation: Training and public relations

Tasks Prepare staff at all levels in all areas
Prepare user community

Costs Vendor supplied training
Staff time to train
Staff time to be trained
Equipment
PR campaign
Duplication, postage, telephone, etc.

Phase 5a / System costs: Hardware

Tasks Install and maintain system hardware

Costs CPU
Tape drive
Disk drive(s)
Terminals
Optical input devices
Printer(s)
Communication equipment
Interface equipment
Installation and shipping

Maintenance contract

Phase 5b / System costs: Software

Tasks Install and maintain system software

Costs Operating system software
Application software
Interfacing software (if appropriate)
Teleprocessing software (if appropriate)
Diagnostic equipment

Phase 5c / System costs: Telecommunications

Tasks Provide links from computer to attachments if necessary

Costs Modems, acoustic couplers, etc.
Auto-answer/auto-disconnect equipment
Cables, connections, etc.

Phase 5d / System costs: Testing

Tasks Ascertain that the system performs as specified

Costs Project manager time
Middle management staff time
Clerical & supporting staff time
Forms, notebooks, etc.
Telephone, postage, etc.
Legal assistance

Phase 5e / System costs: Operation

Task Start and continue system operation

Cost Project manager time
Operation staff time
Middle management staff time
Clerical & support staff time

Patron file conversion
Circulation file conversion
Labeling of collection
Documentation
Supplies (tapes, disk packs, labels, etc.)

Phase 6 / Future development

Tasks Automate other library functions
Plan for equipment replacement
Plan for software revision

Costs Upper & middle management staff time
Equipment
Other

Turnkey Systems

All computer systems must be adapted to the tasks they are to serve and some tasks must be altered to meet the constraints imposed by computers. Technologies which offer varying capabilities and trade-offs do exist. It is important to realize that automated library systems vary greatly in terms of several important characteristics, including:

1) Technology (whether batch, stand alone or distributed)
2) Approach
3) Computer (whether dedicated to one or several libraries)
4) Computer size
5) Function (whether single function or multiple function)
6) Software source (whether in-house, transfer or turnkey)

It became readily apparent to the UTC Library that turnkey systems offered many important advantages over the other software sources available. Development costs of the system software, which are substantial, could be amortized to all customers of a particular vendor and a vendor could thus offer a system at a price that was well within reach of many libraries. Moreover, such a system could be modified by a vendor for a particular library's own environment and because there are several customers with very similiar products, vendors could also offer economical service and support.

The UTC Library found that the available turnkey systems varied greatly and that these variations had significant service ramifications. Such differences included:

1) Programming language
2) Type of processing
3) Equipment
4) Maintenance (whether by one or more organizations)
5) Data stored (whether a full MARC or abbreviated record)
6) How data was stored (variable or fixed fields)
7) Number and type of library functions available
8) Degree of computer sophistication required for operation
9) Other

Selection Process

With all these considerations in mind, the UTC Library issued a Request for Proposal (RFP) for an online circulation control system in August 1982. This RFP contained fifty pages of detailed specifications and related information for the acquisition, installation and maintenance of an automated circulation control system for UTC. Vendors were given 2 months to respond.

Upon receipt of all proposals from qualified vendors, the UTC Library committee carefully evaluated each response according to a predesigned selection process which included (1) a detailed cost analysis of the initial and ongoing costs; (2) a checklist of minimum requirements; and (3) point assignment sheets filled out by each of the committee's members in the areas of hardware, software, system operation and vendor responsibility. This evaluation process lasted approximately eight weeks and resulted in the choice by UTC of a system which features hardware from Hewlett-Packard and a software package developed by Virginia Polytechnic and State University.

The System

The Virginia Tech Library System (VTLS) is designed to be an online, comprehensive, and integrated library system. VTLS not only automates the traditional library services, such as circulation, but also replaces the card catalog and most other manual data files. The system has been operational in its developmental form since 1976 and is currently installed in several locations in the United States and abroad. The salient features of the latest release of the system include:

1) A comprehensive circulation control module;
2) Full online capability to retrieve and modify all data;
3) The ability to handle full MARC records or abbreviated records;
4) The ability to store, retrieve, and edit serial holdings records;
5) The ability to make global changes to author and subject entries;
6) User-friendly screen design based on the concept of screen networking;
7) Easy-to-read screens;
8) Ability to support multiple libraries with one or more computers.

Features which are currently under development include: coded holdings, serials receiving and claiming, acquisitions and fund accounting, additional management reports, comprehensive (MARC-based) authority control, and reserve reading room control.

The UTC Library's hardware configuration includes one HP3000 Series 42 with 2 Mb of main memory, one 7976 high speed tape drive, three 7933 disc drives, one 2608 high speed line printer, eleven CRT's (five with barcode readers), one 2635 printing terminal, one remote support modem and one HP 125 microcomputer with a 2601 printer. The UTC Library's host computer is located in the University's Computer Center and is connected to the Library's peripherals via fiber optics cabling.

The system utilizes barcode labels affixed each circulating item in the Library's collection and to each library patron's i.d. card as a means to link circulation records to book and patron information. A label is an adhesive-backed, machine-readable strip imprinted with a barcode number. Barcoding is a method of representing numeric or character information with combinations of printed bars of varying thicknesses. Each label in the UTC Library system also contains eye-readable characters so as to identify the machine-readable numbers.

Labels in the UTC system are either item or patron labels. Patron labels are produced on site with a barcode printer and simply contain a patron's social security number in machine-readable form. Item labels are produced off site and contain a 14-digit random number. The first digit of this number indicates to the system the category of label (i.e. either item or patron). The second through fifth digits form a four digit library identification code, potentially useful in distinguishing books which belong to other libraries. The next eight digits represent a sequential number that is randomly assigned and unique to each item. The last digit is a check digit.

In order to check out items from the library, patrons are no longer required to sign a item card for every item which they wish to borrow. Instead, Library staff simply use a barcode reader and light pen to scan the labels on the items and on the patron's i.d. card. The patron's Social Security Number and the item's unique barcode number are then linked and the circulation transaction is complete. The system is programmed to prevent circulation whenever certain problems exist, such as overdue fines, and to automatically produce many of the notices and communications which were once produced by hand. In this way the system has met the UTC Library's goal of reducing the amount of patron and staff time involved in the completion of circulation tasks and of increasing the efficiency and accuracy with which these tasks are carried out.

Inventory Control in the Educational Institution

Greg Bean, Systems Manager
Northwest Local School District
Cincinnati, Ohio 45239
Phone (513) 923-1000 ext. 76

Introduction

Although accounting for an educational institution's fixed assets (furniture, fixtures, equipment) parallels that for those of commercial organizations, there are significant differences. "Fixed asset" packages written for industry highlight such features as book and tax depreciation methods, investment tax credit and recapture capabilities; "inventory control" business software is written either for the wholesale distributor or the manufacturer. These differences can make the available, "off the shelf" fixed asset or inventory control packages an ineffective and uneconomical alternative for the educational user.

This article will address the design and implementation of an inventory system that enables an educational organization using capital assets to make more effective use of, and exercise better accounting control over, such assets. The emphasis will be on the implementation process, for the complexities found there will present more of a challenge to the project manager than the development of the software .

In the true spirit of providing support to user departments, I set out to research existing thought on inventory control policies and procedures. The tough part, as with many systems, is not to create that "glorified file cabinet" called inventory control software, but rather to provide the environment that would allow the software to perform successfully. Therefore, this article might be better entitled: A Practical Guide for the Educational Administrator - Implementing Inventory Control.

What Is Inventory Control ?

Inventory Control = Property Control = Fixed Asset Accounting = Equipment Accounting.

Consider, if you will, that each of these labels is interchangeable. What I call Inventory Control you might call Fixed Asset Accounting.

For the purpose of definition the inventory control system described in this article accounts for "equipment" rather than "supply" or expendable items. Ohio's State Auditor specifies the following criteria for items to be classified as "equipment":[1]

An equipment item is a material unit which meets all of the following conditions:

1. It retains its original shape and appearance with use.
2. It is nonexpendable. That is, if the article is damaged or some of its parts are lost or worn out, it is usually more feasible to repair it rather than replace it with an entirely new unit (which is not true with supplies).
3. It represents an investment of money which makes it feasible and advisable to capitalize the item.
4. It does not lose its identity through incorporation into a different or more complex unit or substance.

The State Auditor provides guidelines in the form of an alphabetical listing representing many of the common equipment items. Examples are computers, desks, automobiles, typewriters.

Why Automate Inventory ?

The reasons and benefits for an inventory control system will vary in importance from one institution to another. However, each organization, large or small, university or K-12 districts, will accrue a number of benefits from an effective inventory control system.

- Reduced expenditures.

  An accurate and timely inventory system will show what assets are in use, the condition of those items, and the expected life of the equipment. With this information, more knowledgeable decisions can be made which will avoid duplication of equipment and other unnecessary purchases.

- Meeting regulatory requirements.

  Often publicly controlled institutions are directed by statutory or other regulations to adequately manage institution property. An automated inventory system can fulfill such regulatory requirements.

- Reduced losses.

  A sound system uncovers disappearance, theft, and unauthorized movement of equipment.

- Insurance coverage.

  Information from the system helps avoid the hazard of too little insurance and the waste of too much. Additionally, an accurate and timely system provides equitable and prompt insurance settlements.

- Project / program cost analysis.

  The system assists in identifying property acquired under sponsored research projects or government programs whether such property is owned by the institution, by the government, or by another sponsoring organization.

- Operational efficiency.

  Inventory reporting provides the data necessary for effective management of resources. Equipment can be transferred to provide the most efficient use of the asset and unnecessary items can be eliminated.

System Description

Although the structure of an Inventory data base will depend on the needs of each institution, the following data items are found in most systems:

Inventory number
Description
Building
Location
Area
Equipment type
Serial number
Condition
Life
Acquisition code
Acquisition date
Acquisition cost
Replacement cost
Purchase order number
Account number
State/Federal number
Disposition code
Disposition date

Inventory number - Institution assigned number from affixed identification plate or tag.

Description - Physical description and/or manufacturer's model number/style.

Building - Building number where the item is located.

Location - Room number or office designation where equipment is located.

Area - Individual or department responsible for the item. For example the treasurer may be responsible for equipment in a number of rooms or offices.

Equipment type - Equipment may be classified in different groups (e.g. audio-visual, computers).

Serial number - Manufacturer's identification number.

Condition - The item's physical condition.

Life - anticipated years of use for the item.

Acquisition code - Source of equipment acquisition (e.g. purchased new, used, gift, etc.).

Acquisition date - Purchase date of equipment as per invoice, date of receipt, or purchase order.

Acquisition cost - Purchase cost plus all expenses incidental to acquisition. Assets acquired by gift should be recorded at fair market value plus incidental expenses.

Replacement cost - Market value replacement cost with consideration for physical depreciation, additions, and betterments to the asset.

Purchase order number - Purchase order number that the item was ordered on.

Account number - Account number that was charged with the acquisition of the item.

State/Federal number - Inventory number that is assigned by a State or Federal program/project.

Disposition code - Coding for item to explain the item's retirement (e.g., sold, junked, lost, stolen).

Disposition date - Date when the item was removed from use.

Transaction Processing

Data is entered into the system via a VPLUS data entry screen. Transactions basically fall into three categories:

- Acquisitions (adds)
- Transfers (changes)
- Dispositions (retirement of assets)

Data is collected in a batch file and processed to verify valid account numbers, building numbers (both against the accounting data base), equipment class, etc. and an audit report is printed.

Reports

The following reports are generated by the inventory control system:

- Inventory master list
- options:
  - by building
  - by ID number
  - by account number
  - by equipment type
- Physical inventory report
- a turnaround document to facilitate the count process.

- Transaction audit report
- acquisitions, transfers, and retirements are entered via the CRT and then listed in report form.

Implementation

Proper implementation of an inventory control system requires the cooperation (both initially and on an on going basis) of several functional areas within the educational institution. Detailed written procedures must be established and made institution policy. Individuals that are to be affected by these policies must be informed and made familiar with the aims and personnel requirements of the system.

A decision must be made as to what items will be inventoried. There should be a minimum item value established that will qualify an item for entry into the system. The basic principle: "The cutoff point is left to the organization to determine and many may conclude that this cutoff should be fairly high to minimize record keeping. While obviously each organization will have to make its cutoff decision based on its size and extent of fixed asset activity, the authors would think that most organizations would establish a cutoff between \$100 and \$250."[2] Considerable expense and work can be avoided by excluding from the system many low cost items that are normally classified as equipment but that fall below minimum standards set for inclusion in the system.

The procedure for collecting the data and placing it on source documents for data entry will vary depending on the existing (if any) inventory system. The following procedures are suggested approaches to the implementation project:

Historical information on acquisition and cost should be obtained from either purchase orders, invoices, or other applicable documents. On some items, especially older ones, it may become impractical or impossible to devote the time and effort necessary to locate the information; it then becomes necessary to establish valuations by visual appraisal. Once the historical information is completed, the necessary field work can be done to complete each record. The field work should provide pertinent information such as present condition and of course, location. The institution's inventory tag number should be affixed as the field work is being done. During this initial physical inventory other equipment that may not have been identified during the review of purchase orders can be located (e.g. items donated to the organization).

In our case, purchase orders for the current year and the prior years were collected and arranged by building. Data forms were filled out for all items that met the definition of "equipment" as described by the Auditor and which had a value greater than the cut off value. It then became necessary to notify the responsible individual(s) for a given building that an initial physical inventory will be taken. This notification should assure cooperation in the count process and allow ample time "to get the house in order". Having gathered the historical data for a large percentage of the items at the preselected location, we then locate (the detective work) and tag equipment. During this process older items (items purchased more than three years ago) can be appraised, tagged, and the appropriate information can be recorded on a data collection form. Once this initial inventory is complete, information can be entered into the system.

Property Tags

The purpose of an inventory item tag is to assign a control number to the asset. Several types of tags are available from a variety of sources. After surveying the available sources our district chose "AUTOGRAPH" plates manufactured by Metalcraft, Inc. These tags are serial numbered to individual specifications and bear the institution name. Many options are available for "copy" on the tag as well as variations on numbering (prefixes, suffixes, etc.), colors, and letter styling. The major factors in our selection of this type of tag were:

- economy
- permanence (affixed with strong adhesive)
- easy attachment (no screws or drilling holes)
- durability (not a flimsy label or tape; .016 anodized aluminum)

Transactions

Policies regarding acquisition, movement, and disposition should be clearly defined. The following matters should be addressed:

- Acquisition

  Practice varies as to whether the system is updated as the transaction occurs, or periodically during the year or at year end. There should be an established, workable method which provides for updating prior to audits and statement preparation.

- Transfers

  Items should not be moved from one location (policy should further define whether location means room or building, etc.) to another without formal written authorization by the proper authority. A source document should be created for use in transfers of equipment detailing specifics of the transaction (transfer to, transfer from, approved by, etc.).

- Disposition

  The asset considered for sale, retirement, or replacement should be evaluated for disposal by the proper authority and arrangements made for the physical removal of the item. A form should be developed to report items that become unserviceable or obsolete or that are no longer needed by departments and offices.

- Physical inventory

  The theory: if assets are properly recorded when acquired, and if retirements and transfers are accurately recorded, the inventory, as determined by the physical inventory procedure should equal the amounts shown by the records. A physical inventory report ("turn-around" document) should be distributed to all responsible departments so that they might identify and count items in their jurisdiction. The report can then be returned indicating discrepencies. All variations should be investigated.

Conclusion

Whether an inventory package is purchased "off the shelf" or developed in-house, the effectiveness of the institution's policies and procedures as they relate to implementation and on going use will ultimately determine the success of the system.

References:

1. **Auditor, State of Ohio.** *Uniform School Accounting System User Manual,* Columbus, Ohio, 1983. p. D-16 to D-22.

2. Gross, Malvern J. *Principles of Accounting and Financial Reporting for Nonprofit Organizations.* Canada: John Wiley & Sons, Inc., 1979. p. 208-209.

Lynn, Edward S. *Fund Accounting Theory and Practice,* Englewood Cliffs, N.J.: Prentice Hall, Inc., 1974.

Metalcraft Inc., 149 4th Street, S.W., Mason City, Iowa 50401. *How to Plan a Profitable Property Control Program.*

JOURNAL
OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

The Computer and the Critic or Computer Support for the Analysis of Literary Texts

Clay C. Ross, Jr.,
Professor of Mathematics and Computer Science
and
Jacqueline T. Schaefer,
Professor of French and Comparative Literature

The University of the South
Sewanee, Tennessee 37375
Phone (615) 598-5931

The assistance of the computer is particularly welcome in studies dealing with extensive bodies of texts and wherever exhaustive vocabulary inventory or systematic comparison is in order. At the University of the South, the system for support of text analysis consists of seven program packages:

- I KWPREP Prepares, edits, and lists source texts
- II FREQ Lists vocabulary in decreasing order of frequency
- III KWIC Generates and prints concordances and vocabulary
- IV SORT Selects and prints specialized concordances
- V MAP Highlights selected words in a text
- VI KWCOMP Compares vocabulary in two texts
- VII GRAPH Uses plotter to produce bar graph of word occurrences

These packages were written during the years 1974-1976 in interpreted BASIC on the HP2000F. In 1980 they underwent a slight revision to make them work more efficiently by taking advantage of new features available in the BASIC on the HP3000 series III. Then KWPREP and KWIC were compiled. This provided a dramatic increase in speed during concordance production and printing. Most valuable in this respect were POS for locating delimiters in the text and ADVANCE and UPDATE for writing the supporting files.

The information obtained is essentially of two kinds, quantitative [FREQ, KWCOMP, GRAPH] and contextual [KWIC, SORT, MAP], with output in several forms, of which the most useful were: two vocabulary lists with corresponding frequency counts, one arranged by decreasing order of frequency and one alphabetically; concordances, paginated, showing keyword, context, and location.

The design was deliberately kept simple, but the operation of and sample applications of the programs may be of interest:

I KWPREP. The source file produced by this program is a BASIC formatted file consisting of a set of sequentially numbered text units stored one per record as a number and one or more strings. The number is typically a line number, and a text unit might be a line or sentence or short paragraph -- so long as the line number and text unit would both fit in 256 words (the 2000F record size).

II FREQ. This program simply takes the vocabulary and frequency counts, sorts in decreasing order of frequency, and then provides a paginated listing of words, frequencies, and relative frequencies.

A listing of the words by decreasing frequency of occurrence reveals immediately the relative presence of the structural element, or of the concept for which a lexem stands in the text or texts. This objective reordering of the data alerts the critic to the salient concerns of the text revealed by the concentration of the vocabulary.[1]

An alphabetical listing provides easy reference to the frequency count of any lexem, or lexical unit, and of any family of lexems by simple addition. For instance, the program has been applied to names and denominations of protagonists in a narrative, in order to assess the relative importance of their various roles. Also it has been applied to the analysis of the distribution of personal pronouns; (1) the self-centered first person singular as opposed to the unifying first person plural; or (2), in some languages, the balance between formal or intimate second person; all of which yielded crucial information on the protagonists' attitudes and relationships.

However, one cannot insist enough on the fact that, except in the dictionary, lexems carry meaning only in relation to their context. The FREQ results must therefore be used always in conjunction with a concordance produced by KWIC. The concordance also provides corrective factors for the quantitative data which may be necessary since the computer lists under the same heading words, graphically identical, but semantically different.

III KWIC. This large package contains several subprograms and manipulates four BASIC formatted files having these standardized names and the indicated functions:

<text> The source file described above

<text>K The primary intermediate output, consisting of key word, number of occurrences, initial location, pointer to duplicates in <text>D, left link, and right link, with each record containing a block of 10.

<text>D A collection of linked lists whose heads are the pointers in <text>K and whose entries are [location, pointer to successor], blocked 43 per record.

<text>C A file containing 0 or more words which are not to be processed as key words. (The COMMON words.)

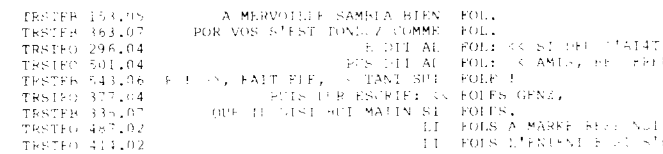
Upon first running KWIC on a particular text one requests that the actual location and sorting of keywords be done. When this is completed, and on any subsequent run, the vocabulary can be listed or all or part of the concordance can be produced. Available selection criteria for producing partial concordances are:

- A particular word
- Any word which lexically follows a given word or syllable
- Any word which lexically precedes a given word or syllable
- Any word which contains a particular syllable
- Any word which terminates with a given letter or syllable

As has been noted[2] a concordance "can and should give rise to new and multiple readings of the text." The context of any given lexem can be read vertically to reveal its paradigm, or horizontally to identify its syntagmatic function. The definition of a term may be attempted, specific to a text or to the author, if the author used the same word in other texts.

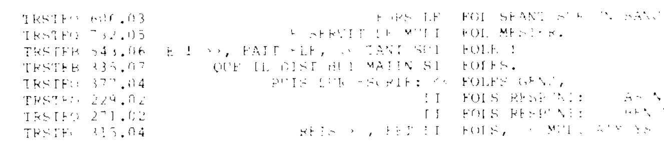
Partial concordances find multiple uses such as the analysis of rhyme patterns, alliterations, and assonances, verb tenses, or for instance case endings when dealing with inflected languages.

IV SORT. This is a collection of programs chained together which allow selection of a set of words from the key word files of one or more texts. These words are then used to produce essentially a combined concordance from the various texts, in the sense that each occurrence of every selected word is written to a file, along with its location in the text, left context, right context, and text identifier. Once done, these five-field entries can be sorted as often as desired on any set of fields, and the resulting special concordance printed. When no more sorting is desired for this set of words, a new set can be selected and the process repeated as often as desired.



```
TRSTFB 153.05          A MERVOILLE SAMBLA BIEN  FOL.
TRSTFB 363.07      POR VOS S'EST [illegible] COMME  FOL.
TRSTFO 296.04                        E DIT AL  FOL: << SI [illegible]
TRSTFO 501.04                       [illegible] AL  FOL: << AMIS, [illegible]
TRSTFB 543.06  [illegible], FAIT [illegible], [illegible] TANT SUI  FOLE !
TRSTFO 377.04          PUIS [illegible] ESCRIE: << FOLES GENZ,
TRSTFB 335.07     QUE IL [illegible] MATIN SI  FOLES.
TRSTFO 487.02                             LI  FOLS A MARKE [illegible]
TRSTFO 414.02                             LI  FOLS [illegible]
```

Figure 1. Fragment of a partial concordance of the semantic field of 'FOL' (fool) alphabetized by right context within keyword. The language illustrated in all figures is mediaeval French, and the texts are the Berne (TRSTFB) and Oxford (TRSTFO) versions of the *Folie Tristan* poems.

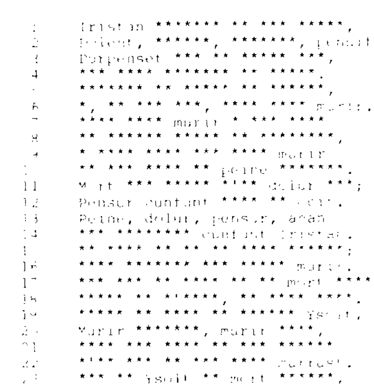


```
TRSTFO [illegible].03                    [illegible] LE  FOL SEANT [illegible]
TRSTFO [illegible].05          [illegible] SERVIT LE [illegible]  FOL MES[illegible].
TRSTFB 543.06  [illegible], FAIT [illegible], [illegible] TANT SUI  FOLE !
TRSTFB 335.07     QUE IL DIST [illegible] MATIN SI  FOLES.
TRSTFO 377.04          PUIS [illegible] ESCRIE: << FOLES GENZ,
TRSTFO 229.02                             LI  FOLS RESP[illegible]
TRSTFO 271.02                             LI  FOLS RESP[illegible]
TRSTFB 315.04            [illegible], [illegible] LI  FOLS, [illegible]
```

Figure 2. Fragment of a partial concordance of the semantic field of 'FOL' ordered by line reference within keyword.

Smaller concordances of lexical, semantic or conceptual networks, be they unique to one text or common to several, allow the demarcation of conceptual fields. If the words are retrieved in the order in which they appear in the texts, the career of these concepts can be traced, separately as well as in parallel or antithetical combinations, through an entire text, or several texts simultaneously.

V MAP. This program accepts, as input, a source file of a text and a list of words. It then lists the text line by line so that all words not in the specified list of words are replaced by ***. . . *, thus highlighting the chosen words which appear unmodified. This offers the visual representation of the presence of a word, or a set of words in an entire text.



```
 1   Tristan ******* ** *** *****,
 2   [illegible], ******, *******, [illegible].
 3   Purpenset *** ** ***** ***,
 4   *** **** ******* ** *****.
 5   ******* ** ***** ** ******,
 6   *, ** *** ***, **** **** [illegible].
 7   **** **** murir * *** ****
 8   ** ****** ***** ** ********,
 9   * **** **** *** **** morir
10   ** *** **** ** peine *******.
11   Mort *** ***** **** dolur ***;
12   Pensur cuntant **** ** [illegible].
13   Peine, dolur, pensur, [illegible]
14   *** ******** cunfunt Tristan.
15   ** **** ** ** ** **** ******;
16   **** ******* *** ***** murir.
17   *** *** ** **** ** ** mort ****,
18   ***** ** ******, ** **** ****.
19   ***** ** **** ** ****** Ysolt,
20   Murir ******, murir ****,
21   **** *** **** ** *** ******
22   **** *** ** *** **** [illegible].
23   *** ** Ysolt ** mort ******,
24   ****** **** ***** ** [illegible].
```

Figure 3. MAP of the opening 24 lines of the Oxford *Folie Tristan* illustrating the heroes' plight and Tristan's suffering, mental agony, and wish for death.

VI KWCOMP. Compares two key word files <text1>K and <text2>K produced from different texts by listing alphabetically and counting words which are

- unique to the first text
- common to both texts
- unique to the second text

A summary of the various totals is provided.

	TRSTFO		COMMON		TRSTFB
		108	FOL		
				369	FOLAGE
				370	FOLE
		109	FOLES		
493	FOLIF				
				371	FOLOR
494	FOLS				
				372	FONDRE
				373	FONZ
495	FORCES				
		110	FOREST		
				374	FORMANT
		111	FORS		
496	FORT				
497	FORTF				
498	FOU4N				
				375	FOUS

Figure 4. This is a fragment of the semantic field of 'FOL'. It was produced by VOCOMP.

Examples of the use of this feature include the comparison of different renderings of a common basic text by different authors; of the successive revised versions of one text by its author; and of similar, but not identical, formulations of the same topic.

VII GRAPH. This program operates on a text file and a collection of words to produce a bar graph on a standard sheet of paper. The text lines appear from left to right with the bar height at each line being proportional to the number of occurrences of members of the word collection which occur on the line. The base line of the plot is always 10 inches, so several graphs can be compared by drawing them on the same sheet using different colors. When they are gathered in one diagram, comparison of word location helps us understand the dynamics of the text(s).

The complete package of programs has been used for research on two mediaeval poems which develop the same episode of the Tristan legend and whose different tonalities had been judged so far exclusively by criteria external to the texts.[3]

It has proved both useful and exciting, in various literature courses, for instance to approach reputedly difficult texts such as T. S. Eliot's *Waste Land*, or Vale'ry's "Le Cimetie're marin," or to demonstrate the common characteristics of Baroque poetry across the barriers of national languages. In yet another course, one student project illustrated the estrangement of two heroes of Racine's *Andromaque* by comparing the concordances of their respective speeches, and analysing the specific mental universe which each revealed.

This is, of course, an ongoing project. We plan to add a program for producing lemmatized concordances, in which each lexem will be identified syntactically as verb, noun, adverb, etc. and listed under its *lemma*, i.e. canonical form as it would appear in the dictionary. There its total frequency of use would be given, along with every occurrence of each variant, with its line reference and surrounding context.

Notes.

1. It must be remembered that on all quantitative data, statistical tests have to be applied in order to establish the hypotheses concerning the variance within the individual text, or between several texts. The Pearson chi-square test, with Yates' correction for small numbers, was used for variance between two elements; the analysis of variance, or the F test, was used for variance between groups of more than two elements.

2. Paul Tombeur in *Vox Latina: Belgian Initiatives in Data Processing the Intellectual Languages of Europe, A.D. 197-1965* Computers and the Humanities, 12, No. 1-2 (1972), 15.

3. The results may be read in *Towards a Poetics of the Mediaeval Tristanian Universe: A Computer-Assisted Analysis of the Folie Tristan Poems,* by Jacqueline T. Schaefer in *Tristania*, VI, No. 1 (1980), 3-18 and No. 2 (1981), 3-22.

The SIGI Career Guidance System for the HP3000

Bill Toll, CDP
Associate Professor of Computer Science
Associate Director of Computer Services
Asbury College
Wilmore, Kentucky 40390
Phone (606) 858-3511 X 237

This article will discuss two related topics: the SIGI career guidance system and its conversion to run on the HP3000 computer. SIGI (System of Interactive Guidance and Information) is a computer based aid to career decision making, created and marketed by Educational Testing Services of Princeton, New Jersey. "SIGI" is a registered trademark of ETS.

SIGI is designed for students enrolled in, or about to enter, two or four year colleges. It is currently used at approximately 250 locations. SIGI is not intended to replace counselors, but to improve the quality of the counseling effort.

In the first section the purpose of such systems, their function and the reasons for the selection of the SIGI package will be discussed. Also included will be an overview of the SIGI software and its format as seen by the user.

The second section will discuss the conversion of the package to the HP3000 with the hope that a look at some of the issues involved and problems faced will be of benefit to others who may need to convert software.

SIGI and Career Guidance

College students today are very career oriented. They want to be prepared for a career immediately upon graduation and, therefore, need to have a career goal in mind as they study. A career guidance system should as a minimum provide a wide variety of information about as many careers as possible and the preparation necessary to enter them. It should ideally go beyond merely providing information and should assist in the decision making process.

Asbury College is a small Christian Liberal Arts college with approximately 1125 students. Our students are becoming more concerned about properly preparing for a career; an obligation was felt to assist them as much as possible. Another important issue in the college world of today is student recruitment and retention. The ability to assist students in the important step of career choice was viewed as an asset to our recruitment program. A study of the students who left Asbury College before graduation found that many of them did so at least partially because of a lack of clearly defined goals toward which to work. It would require the retention of only a few students to match the cost of a system which could help meet these goals.

While searching for a career guidance system, several factors were considered. Among these was the ease of use for a student with no prior computer exposure. Other important considerations were the quantity of information available, the manner of presentation and some attempt at providing the student with a way of evaluating the information. We felt that SIGI was the obvious choice for our needs.

In addition to providing information on approximately 200 careers, SIGI provides a very "slick" packaging of its information with effective screen handling. Techniques, such as leaving certain information on the screen while providing auxiliary information in overlays on other areas, are used extensively. The overall appearance is one which keeps the attention of the user without seeming to be busy or distracting. When multiple items are selected from a menu of possibilities, each selection is flagged as it is made. Simple bar graphs are also used to clarify certain points. SIGI is easy to use in that it requires the student to use the keyboard only to input numbers and to advance to the next screen, by pressing the NEXT key (space bar), or to obtain a printout of the screen by pressing the PRINT key (period). All questions are given in multiple choice format. There is no need to use the carriage return.

One of the strongest points in favor of this particular package is that it attempts to force a student to clarify his values and to use these values to narrow the range of potential careers to research. In the process of the selection of careers to study, instruction is given in basic decision-making processes. SIGI does not attempt to make a career choice for anyone, but offers a systematic exploration of career options. Above all, SIGI forces the user to think.

As viewed by the student, SIGI consists of five parts. Normally a student will complete one of these sections in a single session, although the provision is made to exit from the first section and complete it at a later time. A typical user will require 2-3 hours to complete all of SIGI, with this time divided into 5 o[illegible] sessions. There are also programs to provide management functions such as adding and deleting student users and monitoring their progress through the program. The answers to questions and tentative reer decisions are kept confidential.

The first section of SIGI is the VALUES section. Here, the student is requested to rate the importance of ten values (high income, prestige, independence, helping others, security, variety, leadership, working in the main field of interest, leisure, and early entry) on a scale from 1 to 10 with the sum of the rating points being restricted to a total of 40 to force the student to prioritize his values. Possible fields of interest are scientific, manual/technical, administrative, personal contact, verbal and aesthetic. A game is then played in which the student must make a choice from two possible fictional occupations which exhibit certain of the values. Shortcomings of the chosen occupation or advantages of others are mentioned, and the student again needs to choose to retain his present occupation or to trade it for another.

Inconsistencies between these choices and the stated priorities are displayed in the form of a balance. The game continues until SIGI can judge the relationships between all the values. A chance is then given to modify the weights previously assigned to the ten values. These value weights are presented often in other sections as a reminder. The values section seems to be the favorite of most users. It is also the section that makes SIGI unique and more valuable than other computer guidance packages examined.

The LOCATE section of SIGI allows you to choose any five values and rate their importance. For example, a user may require a below average, average, or above average amount of variety. The occupations are rated along the same guidelines and a list of occupations meeting the criteria is given. For example, a desire for the scientific interest field, a great amount of independence, an average amount or more of security, a great amount of opportunity to help others and a lack of concern for leisure results in a list of 6 occupations. This list is repeated later in other sections to serve as a reminder of occupations to investigate. The student may also ask how any given occupation compares to his stated values to determine which criteria caused its rejection. One of the benefits of SIGI is that career possibilities have been evaluated in these categories by experts in the field.

The third section of SIGI is COMPARE in which up to five questions can be asked about any three occupations at a time. There are 28 questions from which to choose including the definition of the occupation, beginning pay, chances to lead, prestige level, variety, national employment outlook and security. The answers to a given question for each of the requested occupations are displayed at the same time for comparison. A reminder list of occupations selected by the LOCATE section is presented although questions can be asked about any occupation. Values are again listed to aid in the choice of questions. Answers to the questions are updated annually by ETS.

The PLANNING section of SIGI allows the examination of preparation programs for the occupations. Entrance requirements and typical college courses are listed. An attempt is made to cause the student to evaluate the risks and his chances of success in preparing for a given career. For instance, the amount of preparation time required is listed and the user is asked if he is willing to make that commitment.

The final section of SIGI is STRATEGY where the rewards and risks are evaluated. An illustration involving the purchase of an automobile is used to aid the student in developing decision-making skills. Louis Logic needs to buy an automobile but none of the available models is perfect, so he must determine his values and their relative importance and discover how each model compares for his set of values. The rating for each value of the occupation being evaluated is multiplied by the student's weight of that value, with the sum of these being defined as the desirability sum. In a similar manner, the student estimates the probability of successfully preparing for, and entering, an occupation. This probability is then used in conjunction with the desirability sum and compared for different careers to aid in making a tentative career choice.

As can be seen from the discussions of the individual sections, the concept of values - the satisfactions desired from an occupation - ties together the sections of SIGI.

A student is required to complete all previous sections of SIGI before preceeding to the next. Once a student has completed all sections, he may redo any of the sections in any order desired. This process allows for review or evaluation of other possible occupations.

The Conversion of SIGI

The choice of SIGI to aid in career guidance on our campus was simple because of the features already mentioned. Unfortunately, there was no version available to run on an HP3000 computer. After some deliberation the decision was made to convert the package. SIGI was originally written in BASIC-PLUS for the DEC PDP-11 computer system, but a FORTRAN version had been written and modified to run on several other machines. ETS recommended that we purchase the PR1ME computer version from the State University of New York at Brockport, which we did. This choice was probably a mistake, in that the PR1ME machine is much like the PDP and very much different from the HP. An IBM version would probably have been a better choice because of more compatibility between FORTRAN versions. The code we received from SUNY-Brockport did not have all the bugs removed as it was a preliminary version, and indeed could not have compiled on any machine because of some basic FORTRAN errors.

The SIGI program is actually an interpreter for a language named RIL which was developed for this purpose. The instruction set of this interpreted machine is very much like that of the PDP and, therefore, is oriented towards its architecture. The RIL pseudo machine has a multiple stack structure and would be worthy of study for its own merit.

There are some advantages to this approach. Significant changes to the SIGI program as viewed by the user can be made by changing the code that is interpreted without changing the interpreter so the programs run by the user do not change. The RIL language was designed to be a good language for the expression of the types of operations necessary for the sophisticated screen manipulations and data substitutions done by SIGI. It is also a good approach in that it provides for easy testing of new additions or modifications.

During the conversion process, a new appreciation was gained for the strong variable-typing characteristics of HP. For instance, subroutines written with one dimensional arrays in mind were passed two dimensional arrays in the code we received. HP FORTRAN requires exact type and dimension matching between parameters and arguments. Such problems and others kept us from using character variables without totally rewriting the package.

Often, the original BASIC-PLUS code was the only way to determine the intent of what was being rewritten. It was tempting to begin from scratch, especially when one more instance of undocumented conversion factors was found in the FORTRAN code. These were a problem because PR1ME assumes the ASCII parity bit to be always 1, where HP assumes a 0, causing the decimal values in the comparison statements to be incorrect because of the leading bit of each halfword. Although the original language for SIGI was BASIC, it would have been extremely difficult to write the package in HP BASIC because the data to be displayed on the screen was manipulated in 1024 byte blocks and HP BASIC character strings are limited to 255 characters.

The package, as received, did have one very nice feature. Various terminals can be handled easily by reading their cursor control sequences from a file which is easily set up during the initialization of SIGI. Terminals without such features as erase-to-end-of-page and erase-to-end-of-line are supported, but their action is too slow to be of practical use. Hewlett-Packard terminals perform very well, especially the 2621P with built-in printer. Also implemented was the capability to do a screen dump to a spooled printer so that other terminals could be used. Because of the sophisticated screen manipulations, SIGI screen I/O is handled by the FREAD and FWRITE intrinsics after proper initialization with FCONTROL, FOPEN and carriage control sequences.

After the code was converted, the next major hurdle was in reading the data files supplied by ETS, including the frames to be displayed and the RIL machine code to be interpreted. ETS followed a pattern seen at other PDP installations - they insist on sending everything out in PDP backup format. I believe they would also find it interesting if they needed to read a STORE tape from an HP shop. This problem was solved with a knowledge of the format used and a few simple programs which found the logical record boundaries, and reblocked the disc file produced by reading the 2 Kbyte record tape. We also discovered that, although their tape was a subset of the ANSI standard for tape labels and HP supports a subset of the same standard, they are apparently different subsets. In the fall of 1982, ETS switched to a VAX system. Now we could recognize the tape as a labelled tape, but could not read it when treated either as labeled or unlabeled. We finally convinced ETS to send a nice simple unblocked, unlabeled ASCII tape. We have provided other users of HP3000 SIGI with store tapes of the converted files.

The completed HP3000 version of SIGI consists of approximately 8600 lines of FORTRAN source and 3600 data records of 1024 bytes each. The data frames are organized so there are a minimum of blanks between information so that this data file represents a large amount of information supplied by ETS. In addition, the RIL code to be interpreted consists of 39 records of 1024 bytes each.

The capability of the HP3000 to allow multiple users to share one code segment is helpful, however each user must have his own data segment which needs to be the system maximum of 32K. At one point, before some unneeded array space was eliminated, an attempt was made to use an extra data segment by splitting a large two dimensional array into 2 pieces, one half in the normal data segment at any time, and the other in the extra data segment. This approach quickly proved to be a disaster because one set of circumstances caused a 1024 byte transfer between the two halves which resulted in 2048 exchanges of the array halves and a rather long wait for the particular display to appear. The data was later trimmed to fit comfortably into one data segment.

The results of this conversion effort was a SIGI package which performed identically to the original and was very helpful in our counseling work. It was, however, almost too slow to be usable. Students became bored waiting for the display to change and two SIGI users on the system brought about a noticeable performance degradation. Our system at the time was a Series III with 1 Mb and a typical load of 12-16 sessions.

Our initial effort was merely to get the code functioning properly in time for the opening of the Fall quarter and had involved the work of two people for about half of the summer of 1981. During the summer of 1982 an effort was made to improve performance with the likely alternative being the purchase of the microcomputer version of SIGI. In what seemed to be just short of a miracle, a revision of the code resulted in CPU times that were less than one-tenth of the time previously required.

To find the bottlenecks we logged each RIL instruction to disk as it was interpreted, along with its execution time, and wrote a program to analyze this data. Not surprisingly, we found that I/O was done often. The major activity of the program is finding and displaying the proper information with the rate of display being dependent upon the student. Upon inspection, it was discovered that data was read in packed format, unpacked for comparisons and searching, and repacked for output. Reading and writing in unpacked format resulted in a significant saving of CPU resources.

Although the codes involved for searching the display files for data and substitution information for the correct occupation were executed relatively rarely, they consumed a significant fraction of the CPU time. Upon study of the code involved it became clear that two things, which should not happen, were being done. First, the searching was being done in a strictly linear fashion. A record was read and searched and if the data needed was not found, the next record was read and searched. Typically, one would be searching for information for 3 or more occupations which were not entered in order and were not sorted prior to the search. Therefore, the search began at the beginning for each item. To remedy both problems, a program was written to build an index file so that direct reads could be used to find the proper record. Before this improvement, after occupations matching the values were found in LOCATE, several minutes could be spent waiting for the internal occupation numbers to be converted to corresponding occupation names. Now this display appears almost as fast as it can be printed.

Summary

SIGI has been found to be an effective aid to career counseling. As distributed by ETS, it contains much information which is presented in a very appealing and sophisticated way. In the process of conversion of this package to run on the HP3000, much has been learned which has been of value in other conversion efforts.

SIGI has made a very favorable impression at Asbury College on both the counseling staff and students. At present we are anticipating its use by all new students during the 1983-84 school year.

Following this article are typical SIGI displays from the various sections. These displays are copyrighted by Educational Testing Services.

Further information on SIGI can be obtained from Educational Testing Service, Princeton, NJ 08540. Information on the HP3000 conversion is available from the author.

```
      VALUES                   WEIGHT  (Higher number = more importance)
                               0         2         4         6         8
                               '----'----'----'----'----'----'----'----'
   (1) HIGH INCOME             5----*----*----*----*----*
   (2) PRESTIGE                1----*
   (3) INDEPENDENCE            5----*----*----*----*----*
   (4) HELPING OTHERS          4----*----*----*----*
   (5) SECURITY                5----*----*----*----*----*
   (6) VARIETY                 3----*----*----*
   (7) LEADERSHIP              3----*----*----*
   (8) INTEREST FIELD          8----*----*----*----*----*----*----*----*
***(9) LEISURE                 2----*----*
   (0) EARLY ENTRY             2----*----*
                              ---
                       SUM = 38

LEISURE: Short hours, long vacations, or the chance to choose your
         time off.  Most important satisfactions are off the job; so
         work must not interfere.

   To ADD a point to this value, press the number 1.
   To SUBTRACT a point from this value, press the number 0.

When you have added or subtracted as many points as you want, press
   NEXT.

                       MODIFYING VALUE WEIGHTS

*This display material copyrighted by Educational Testing Services
```

```
Press one of the numbers (1-6) to show which field listed below is
most interesting to you.  The field you chose earlier is marked with
***.  You can choose it again if you still like it most.

***(1) SCIENTIFIC--data, experiments, analysis, mathematics
           Examples: chemist, botanist, economist, physics teacher.

   (2) MANUAL/TECHNICAL--things, machines, tools, electronics,
                         mechanics
           Examples: welder, mechanic, technician.

   (3) ADMINISTRATIVE--business, finance, records, systems,
                       management
           Examples: accountant, secretary, hospital administrator.

   (4) PERSONAL CONTACT--people, serving, selling, advising, teaching
           Examples: nurse, social worker, sales worker, teacher.

   (5) VERBAL--words, writing, public speaking, language
           Example: news reporter, clergy, English teacher.

   (6) AESTHETIC--art, design, music, dance
           Examples: artist, singer, designer, art teacher, dancer.

           CHOOSING AN INTEREST FIELD IN VALUES SECTION

*This display material copyrighted by Educational Testing Services
```

Welcome to the Strive Employment Agency.

The jobs currently available are listed below.

(1)

ORTHOLOGIST

This is a wonderful job if you like activities in the scientific field of interest. You spend most of your time working with scientists. You help them make observations (sometimes on fantastic instruments), collect data, and calculate and analyze the results of their work.

(2)

MOGULIST

This job, Mogulist, gives you a lot of prestige in your community. Because you're a Mogulist, clerks will be courteous in stores, banks will cash your checks instantly, police will overlook minor traffic offenses, your name will be in the newspaper in a nice way, and you will be asked to serve in fund drives and on advisory committees.

Press (1) or (2) to show which job you prefer.

TWO OF THE FICTIONAL JOBS IN VALUES GAME

*This display material copyrighted by Educational Testing Services

Values for locating occupations:

(1) Interest Field
SCIENTIFIC

(2) Independence
A great amount

(3) Security
An average amount

(4) Help Others
A great amount

(5) Leisure
Don't care. A small amount OK.

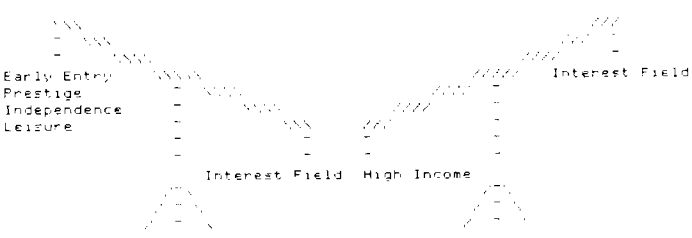
These occupations meet your specifications

129 Dentist
159 Physician
205 Speech Pathologist/Audiologist
236 Optometrist
270 Psychologist, Clinical/Counseling
298 Podiatrist

All occupations fit this spec. Press PRINT (for a copy) or NEXT.

LIST OF JOBS MEETING SELECTION CRITERIA

*This display material copyrighted by Educational Testing Services



Look at the balance scales above. Your choices in the game suggest that Interest Field was more important to you than any Value on the left side of the scales.

Now look at the balance scales above. Your choices in the game suggest that Interest Field was LESS important to you than the Value on the left side of the scales.

Press NEXT.

RESULT OF CHANGING JOBS IN VALUES GAME

*This display material copyrighted by Educational Testing Services

OCCUPATIONS	DESIRABILITY SUMS	CHANCES
129 Dentist	144	[illegible]
159 Physician	147	80
270 Psychologist, Clinical/Counseling	141	50

You have chosen 159 Physician.

This occupation looks like a good choice. Of the three occupations on the screen, it has the highest Desirability Sum [illegible] within ten points of the highest.

Also, according to your estimate, your chances of getting into it are greater than or equal to your chances of getting into either of the other two occupations. It is the best with respect to risk.

Thus it combines the most reward with the smallest risk, and you should choose it, as you did, according to Rule 2 of your strategy.

For a copy of this display, press PRINT. Otherwise press NEXT.

RESULTS OF STRATEGY SECTION

*This display material copyrighted by Educational Testing Services

Values for locating occupations:

(1) Interest Field
SCIENTIFIC

(2) Independence
A great amount

(3) Security
An average amount

(4) Help Others

(5) Leisure

Specification for HELPING OTHERS:

How much opportunity for Helping others must an occupation have to be included in your list?

(4) A great amount

(3) A more than average amount

(2) An average amount

(1) Don't care. A less than average amount will do. (This specification does not keep any occupation off your list.)

Press the number (4-1) for the specification you choose.

SELECTING VALUES IN LOCATE

*This display material copyrighted by Educational Testing Services

DEFINITION AND DESCRIPTION
(1) Definition of occupation?
(2) Description of work activities?
(3) Level of skill in interacting with data, people, things?
(4) Where to get more information?

EDUCATION, TRAINING, OTHER REQUIREMENTS
(5) Early Entry: Education required?
(6) Specific occupational training?
(7) Examples of college courses?
(8) Personal qualifications?
(9) Other requirements?

INCOME (National figures)
(10) Beginning pay?
(11) Average pay (half earn more, half less)?
*(12) Top earning possibilities?
(13) How earnings vary?

PERSONAL SATISFACTIONS
*(14) Help Others: Chances to [illegible]
(15) Leadership: Chances [illegible]
(16) Interest Field: Which [illegible]
(17) Prestige level?
(18) Special problems?

CONDITIONS OF WORK
(19) Physical surroundings?
(20) Leisure: hours, vacations?
(21) Independence on the job?
*(22) Variety?
(23) Fringe benefits?

OPPORTUNITIES AND OUTLOOK
(24) National employment [illegible]
(25) Where are the jobs [illegible]
(26) Security in the [illegible]
(27) Advancement?
(28) How many women?

If you do NOT want to ask any more questions right now, [illegible]
If you want to select another question, press its number. [illegible] here:
If it's wrong, press RUBOUT or DELETE and start over. If it's [illegible] press NEXT.

QUESTIONS AVAILABLE IN COMPARE

*This display material copyrighted by Educational Testing Services

Top earning possibilities (1981-82)?

270 Psychologist, Clinical/Counseling

Best estimate: college (9-10 mo. contract), $34,000-$43,000. Top 16% of fed. gov't psychologists earn $45,800-$50,000/yr.

129 Dentist

Best estimate: many general practitioners and specialists in independent practice earn over $80,000 per year.

159 Physician

About 10% of office-based M.D.'s earn over $150,000 per year.

TYPICAL ANSWERS FROM COMPARE

270 Psychologist, Clinical/Counseling

You have chosen a professional occupation. That means you must get a bachelor's degree and then go to graduate school. If you start at a community college, you will have to transfer.

The total amount of time you have to spend in preparation varies from occupation to occupation. The minimum is FOUR years of college plus ONE year of graduate school. For most occupations graduate school will take TWO or THREE years, and often more. A later display will show you the full educational requirements.

So the first decision you have to make is this: Are you willing to invest a large chunk of time in preparing for this occupation?

(1) Yes, willing to spend the time.

(2) No, not willing to spend the time. I want to switch to another occupation.

PLANNING SECTION

270 Psychologist, Clinical/Counseling

You must have a master's degree or Ph.D. to become a professional psychologist (with only a bachelor's degree your opportunities will be limited). For best preparation:

1. Enroll in psychology in college.
2. If you start at a community college, take courses required for transfer after 1 or 2 years to a 4-year college with a bachelor's program in psychology.
3. Get bachelor's degree.
4. Select a graduate school with a program in clinical or counseling psychology. Take tests required for admission. Both master's programs are available; most psychologists have the Ph.D. Admission may be very competitive.
5. If you plan to enter private practice, you will need a state license and a Ph.D. (in most states). School psychologists in public schools must be certified. CHECK STATE LICENSING AND CERTIFICATION REQUIREMENTS.

Press PRINT or NEXT

PLANNING SECTION

JOURNAL OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

Computing at Purdue University's Schools of Management.

Ben L. Mays
Director of Technical Services,
Purdue University
School and Graduate School of Management.
West Lafayette, Indiana 47907
Phone (317)494-4534

The School of Management and Graduate School of Management are located in the Krannert Building and the Krannert Center for Executive Education on Purdue's campus in West Lafayette, Indiana. The schools have an enrollment of about 2400 undergraduates, 250 students in the Professional Masters Program and another 100 in Doctoral programs. There are around 80 full time faculty and another 100 visiting faculty, lecturers, and graduate instructors. Degree programs include Bachelor of Science in Industrial Management (B.S.I.M.), Bachelor of Science (B.S.) with a concentration in management, accounting, or economics, Master of Science in Industrial Administration (M.S.I.A.), Master of Science in Industrial Relations (M.S.I.R.), Master of Science in Management (M.S.M.), Master of Arts in Teaching (M.A.T.) in economics, and Ph.D. degrees in management, economics, or organizational behavior.

The schools are relatively new on campus, having developed from a need for further management training for engineering majors and others with backgrounds in the various fields of science. The undergraduate school was established in 1958 and the graduate school in 1962. Their academic computing activity, until 1982, was concentrated in the central computing facility on campus, Purdue University's Computing Center (P.U.C.C.) which provides support for academic computing for several schools of the university. P.U.C.C. employed three Control Data Corporation computers (6500 and 6600 with Modcomp front end processors) which have been additionally implemented by Digital Equipment Corporation 11/70 and 11/780 systems and (within the past year) a Control Data Corporation CYBER 205 system. During the initial years, academic computing was done on a punched-card batch mode basis. The punched-card operation is being phased out in favor of video terminals, which are interactive in terms of setting up a program, but computing itself is still batch mode.

In late 1981, in conjunction with Hewlett-Packard, an interactive mode Hewlett-Packard System 3000, Series 44, computer was installed in the Krannert building. The system at that time consisted of the C.P.U., 1.5 Mb. of main memory, a 7970E, 1600 b.p.i., tape drive, a 7925M, 120 Mb., disk drive, a 2608A, 400 l.p.m. printer, a 2635B printing terminal, eight 2621B character mode terminals with integral printers, two 2624A block mode terminals, a 2647A intelligent graphics terminal and a 7221C eight-pen plotter. A non-Hewlett-Packard terminal (General Terminal Corporation GT 100A) was also obtained to provide a terminal with video output that could be coupled into large monitors or video projectors for those instances where an instructor or lecturer would want to employ the computer in a classroom, workshop, or seminar setting. The original software consisted of the operating system (MPE IV) which included IMAGE/3000, a database system, KSAM/3000, a keyed sequential access method file system, and V/3000 and VPLUS/3000, batch forms entry, along with Hewlett-Packard's COBOL, FORTRAN, BASIC (interpreter and compiler), SPL/3000 (System Programming Language), DSG/3000 (Decision Support Graphics), and Scientific Library/3000, and a collection of some statistical and error functions.

By January of 1982 (the beginning of the Spring semester) the system was released for use by faculty and students. It was decided to make it available to faculty and graduate students for research, graduate courses, and studies for theses and dissertations and to administrative personnel for various data bases, while keeping undergraduate usage and large statistical and financial packages that lent themselves to batch mode operation on the P.U.C.C. system. Our initial strategy was to build our program toward overcoming the apprehensions of those who were unfamiliar with computing, or who had had problems in working in a batch mode environment. Towards this end we concentrated our programming efforts on developing packages of documentation on various uses of the system, looked into purchasing commercial and educational, financial and statistical packages that would be relevant to the courses being taught, developed and presented workshops for students and faculty and had discussions with business firms and other universities on what they had found to be worthwhile. In a short time we had acquired I.F.P.S. (Interactive Financial Planning System), an extensive financial package from the Execucom Corporation, I.S.E.A. (Interactive Software for Econometric Analysis) from John Eaton of the London School of Business, LINDO (Linear INteractive and Discrete Optimizer), an interactive system for linear programming developed by Linus Schrage of the University of Chicago's Graduate School of Business, QSTAT, an interactive statistical alysis program by Mike Biderman of the University of Tennessee at Chattanooga and programs from the Hewlett-Packard Users Group Contributed Library and System tapes.

The initial use and acceptance of the system was quite successful. It was helped a great deal by the fact that Hewlett-Packard's System 3000 is basically very user friendly. By the beginning of the Fall Semester of 1982, several programs, both administrative and academic, had been developed internally and were being used in coursework and some seminars for executive groups in business firms. By late Fall we had added MM/3000 and PM/3000, Hewlett-Packard manufacturing and production control management programs, PASCAL/3000, HPSLATE, a text editing package, and programs from the Users' Group Library and System 2000 tapes, some modified and brought up to date for our system.

By Spring of 1983 we had around 140 Master students taking courses in which various computer programs on the 3000 were being used, along with several faculty, doctoral students, and administrative personnel using the HP3000. We continued our programs of offering workshops for students and faculty, of generating documentation for the uninitiated and of obtaining programs. The system s enlarged by expanding the memory to 2.0 Mb., connecting in a 404 Mb. second disk drive, Hewlett-Packard type 7933, and modem capability to enable the system to be accessed by phone line from other terminals and personal computers. We also added several software packages including HPDRAW, a text and figure design package, HPEASYCHART, a program to allow the inexperienced user to generate various kinds of charts, OPT/3000, a system optimizing package, DICTIONARY/3000, a data dictionary and directory, REPORT/3000, a report writer for data formatting and reporting, INFORM/3000, an interactive inquiry and report generation facility, TRANSACT/3000, an application development program, TDP/3000, a text and document processor, MATH/3000, a matrix type electronic spreadsheet program, and self-paced tutorial programs including a tutorial on BASIC which was developed internally from one of the Users Group programs.

At that time, construction on the new building for our Krannert Center for Executive Education was nearing completion and our first accelerated Master's Program in Executive Education began. The concept of the program was to have the enrollees, executive personnel from business organizations who wished to obtain a Master's degree through this intensive design, access our computer through telephone dems for the preliminary coursework and interface h our faculty on the program by means of an electronic mail system. They would then come on campus for an intensive two week session, return home and continue the program through the modems, obtain tests, do coursework, and use the electronic mail interactively between themselves and our faculty. After several weeks they would return again to campus for another two week session and then continue on through the modem link. The concept involved developing the computer programs that would provide the necessary coursework structure and the electronic mail system for communication. This was done and we expanded the number of modem inputs into the system to accommodate these new requirements.

In the Fall Semester of 1983 we will have around 170 regular Masters students using the 3000 in several Management, Finance, Accounting, Statistics, Quantitative Methods, and related courses. We will also have about 20 Doctoral and 15 faculty utilizing programs in study and research as well as various administrative programs. Other schools on campus have begun to run programs that are particularly suited for them on this computer. We are in the process of obtaining an Intelligent Port Selector (Micom) to increase the flexibility and system analysis capabilities of the system and plan on adding memory and additional port capacity. Our faculty has developed documentation for their coursework and this effort is continuing with increasing numbers of members becoming involved. We have also generated a considerable number of programs internally for use with the system. We are investigating the use of personal computers, both as standalone and as adjuncts to the 3000, either modem or direct coupled. In this respect, we have acquired some Hewlett-Packard units which we plan on using in an undergraduate class in Accounting this Fall and we also have some IBM and Apple personal computers.

Our experience has been that an interactive computing system, especially one that is user friendly, will be rapidly integrated into an educational structure. Documentation must be straightforward and put together in a manner that can be used by unsophisticated persons (no trivial task!). It is crucial that documentation on internally generated programs be kept complete and up to date, that programmers be given enough time, that the importance of documentation be sufficiently stressed, and that programmers make sure keeping documentation up to date is a basic and routine function. Where at all possible, programmers should be involved in consulting with faculty and students as little as possible. It is very difficult for a programmer to attain any degree of continuity of effort when there are continual interruptions. A high degree of reliability in both system hardware and software performance is vital, especially where the progress of courses is dependent upon electronic linkages.

There are some caveats in the program, of course.

With limited personnel and resources, care has to be taken that nonacademic program development requests be analyzed conservatively to avoid a pattern of frequent system and staff overloading. Programs of all types have to be continually weighed for the benefits being derived against the resources being absorbed to maintain them. User documentation must include fundamental basics (which should be reviewed in workshops) such as how to get on the system, how and where to obtain help and documentation, how to operate the terminals and other equipment needed, how to get material printed and how and where to obtain hard copy, and all of the other little but necessary details which the experienced programmer or user tends to take for granted, but which present potential problems, discouragements, and stumbling blocks to the neophyte. Our experience with transmitting data via telephone company long distance lines has shown that this is something to be approached with a great deal of care and the use of dedicated circuits or networking may definitely have to be considered. Our experience with terminals also has shown that character mode terminals, as opposed to block mode, have very definite limitations and will frequently be unable to be used for applications involving graphics or other block mode situation. Non Hewlett-Packard terminals must also be chosen with care. Some programs (such as MATH/3000) use escape characters peculiar to Hewlett-Packard terminals and will not run on terminals lacking these characteristics. In the case of large screen projection in classroom applications, the number of projectors that can take the nonstandard frequencies being employed in most terminals and display them clearly and distinctly, with enough lighting to allow for note taking, to large groups is very limited, especially where 80 characters per line is being displayed. These are a few of the caveats. There are many.

Future development of our programs will undoubtedly be influenced by the growth of personal computers in the academic environment, the potential networking of various systems in the schools on campus and other systems, the evolution of our executive education programs and the burgeoning computer expertise of students in the primary and secondary schools who are going to be entering the university community in the not too distant future. We are looking at these considerations by planning to add inputs to our network, whether these be terminals or personal computers. We will also be adding more undergraduate capability to our operation, looking into networking options, connecting our Executive Education Center into the system, developing communication interfaces with other computing systems on campus. We also plan on increasing our graphics and word processing potential, adding new software and having the involvement of our faculty grow in the development of courses utilizing computing in coursework, testing, and CAI (Computer Assisted Instructions). To paraphrase an old Chinese saying, I believe we shall be living in interesting times.

Microcomputer Madness

John Beckett
Southern College
Computer Center
Box 370
Collegedale, Tennessee 37315
Phone (615)238-2080

You wouldn't believe the action back in Catacombs #3 these days. We are making the HP3000 do things it didn't know how to handle before, and reducing its workload. The central fact around which this revolves is simple: Microcomputers are finally a practical reality, to the extent that we can even use them in the deepest jungles of Tennessee!

CP/M on the HP3000?

Yes, that's what we have running. It was originally put together by some students at a sister college of ours. They wrote an emulator for the 8080 chip in SPL, then set up additional interface software to emulate a floppy disk controller. After adding some BIOS code, they were able to load CP/M and have it work!

Floppy disks under CP/M appear to the HP3000 as binary files. Each is 2002 sectors long. You may have up to four "drives."

One feature of CP/M is not available: testing "console ready" during processing. That would require No-Wait I/O on the 3000.

How practical is it? That depends on what you want to do. If you have a badly overloaded Series II and you wish to bring up VisiCalc, the answer is "No." Some applications present themselves, however:

- Using the HP3000 as a debugging tool for developing software which will eventually be used on very small microcomputers. You are operating in a familiar environment, with all the tools (text editors, high-speed printers, acres of disc space) you need.
- The HP3000 can become a central node for distributing CP/M software to different kinds of CP/M-based systems. This will take some additional effort to pull off, but we are working on it.
- Limited use in giving students an introduction to CP/M without buying HP 125's for the whole class.
- Supporting software for occasional use, when development costs, otherwise, would be unjustified.
- Hardware support for a parallel-run conversion where you wished to convert an application from a CP/M-based environment to the HP3000 and want everything to work from one terminal.
- Temporary support until you get bigger disks on the CP/M computer.

Performance—aah, that's the question! Yes, it's slow—slower than most microcomputers. The HP3000 is having to emulate every line of assembler code. If on top of that you are running Microsoft BASIC, the difference could be stunning. But the disks should be plenty fast. In an I/O-intensive operation this could help substantially. In a screen-oriented application such as a spreadsheet, forget it.

All of which brings me to another project that may be of interest:

The Master Assembler

A common problem in colleges and universities using the HP3000 is what to do about teaching Assembler language. The HP3000 does not really have a proper assembly language. SPL may be one of the best ideas in a systems language to date, but the competition has not given the concept their vote of confidence. So we must teach our students programming at this level.

We toyed with the idea of using CP/M 3000 to do the job. Performance just didn't seem good enough. So we considered some other options:

1. Getting micros for them all. Disadvantage: $$$
2. Putting a "hook" into the HP 2621 firmware so that we could download programs and run them on the Z-80. Disadvantages: Not enough RAM to do anything worthwhile, no debugger, and no way to get to hard copy. In addition, the ROM space in the HP 2621B is full. We'd have to remove support for some function to add anything else.
3. Writing an emulator to run on the HP3000, configured for the purpose of developing Z-80 software, which could then be downloaded. Disadvantage: loads the HP3000, since it is a new application.

We settled on option 3. Then we got to thinking—why not support more than one CPU chip? Current plans are to handle all commonly used 8-bit chips, including: 6800/6803, 6809, Z8 6502/6510, Z-80, and 8080.

What will be provided is a consistent operating environment with which the students may learn. The assembler is set to the instruction set you want by a simple $CONTROL option. The assembler handles Macros, and will process source files produced by EDITOR and QEDIT. By implication, this means that QEDIT users will not have to leave QEDIT to do an assembly or execution. The interpreter then allows one to use a consistent debugger to work with the code, I/O devices, and calling sequences which are consistent.

The Master Assembler will be available in December of 1983, with evaluation copies forthcoming at an earlier date.

Mu-Net

Another component of our overall project is terminal emulators. We have discovered three trends that have caused us to think about the whole issue of terminals. The result has been a decision: we will never again buy another terminal for our site! The trends are these:

1. Increasing demands of our users for functions, especially CPU-intensive processes such as word processing and spreadsheets.
2. Decreasing cost of microcomputers.
3. Lack'of a decrease in paper costs.

The last trend is important to us, since most of our academic printouts are debugging runs—we can do debugging via CRT's. Mu-Net is a project that involves writing of terminal emulators for microcomputers that we consider especially good values or especially common. At present, the list includes the Kaypro, Apple, TRS-80 Model III/IV, and Commodore 64. We plan to include at least one of HP's microcomputers later on. Functions we are implementing or will soon implement on these computers include:

- Vertical scrolling a la HP "ROLL" keys, of at least ten pages.
- Horizontal scrolling to handle an effective screen width of 255 characters.
- Suppport of the HP3000 "BREAK" function rather than a control-C.
- Emulation of HP's handshaking protocol.
- Where applicable, support of a printer as is done via HP's thermal printer option.

How will this save us paper? In three ways. First, we have noticed that printouts often are obtained simply because 80 columns are not sufficient to look at some parts of a spoolfile. Second, most terminals do not have sufficient memory to handle enough information for even debugging short programs in COBOL. Third, by making student-owned microcomputers able to act as effective workstations, we will be moving programming activity away from printers.

As of this writing I have been using a Kaypro microcomputer as a workstation. This article, for instance, was written using its inherent word processing capability. Impact on the HP3000 was zero. It also performs as a very effective terminal. I am finding myself preferring it because of the expanded scrolling memory and horizontal scrolling capability. I'll never go back willingly to a single-function workstation.

The terminal emulator was first developed in embryonic form on the TRS-80 Model III. It has been rewritten in full-function mode on the Kaypro. It is also being moved to the Apple. The Commodore 64 and full-function version for the TRS-80 Model III remain to be done.

The relationships between these three projects are interesting. We were working on implementing download hooks in the HP 2621 terminals when CP/M 3000 arrived. Working with it convinced us that there was a good idea, but that it didn't meet our local needs. Hence the Master Assembler. MuNet is being developed using the resources made available through the Master Assembler. The resulting emulators are then downloaded to the target microcomputers. This means we do not have to distribute updates of MuNet emulators—we just instruct users to download the latest version!

Availability

CP/M for the HP3000 is currently being marketed. Contact Mr. Waldemar Janke at (205) 883-2931. We plan to market the Master Assembler and MuNet ourselves early in 1984. Contact us for more specifics on features, performance, or price.

Living in a Multimachine Environment

Michael Klein
Loyola University
6363 St. Charles Avenue
New Orleans, Louisiana 70118
Phone (504) 865-3830

Loyola University is a relatively small, Jesuit, Liberal Arts University in New Orleans. Our enrollment is approximately 3500 students. Although we are primarily an undergraduate school, we do offer several Masters and Professional Degrees. We offer three degrees in computer related subjects. In addition, many other programs have varying levels of computer requirements. In terms of our computer requirements, we are probably typical of other schools our size. Where we differ is in the means we have chosen to provide computer support for the University. Several years ago the University decentralized computers by forming two separate organizational entities: the Computer Center and Academic Systems.

The Computer Center is responsible for providing computer support for the administrative operation of the University. The Academic Systems Office is responsible for the academic computer support. To further this separation, the Computer Center Director reports to the Vice President for Business and Finance while I, as Academic Systems Coordinator, report to the Vice President for Academic Affairs. One might think that this would lead to many areas of conflict and possible confusion. However, the Computer Center Director and I have similar philosophies and a good working relationship. We manage to resolve many of the gray areas of responsibility to our mutual advantage.

The Computer Center operates an IBM 370/138 with about 30 terminals. Academic Systems supports an HP3000, a VAX 11/750 and a host of microcomputers, along with about 40 terminals. All three CPU's are located in the same facility, but have no CPU to CPU links. Under normal circumstances, there is no need for the three systems to interact. However, on those rare occasions, we can interchange data between the systems using magnetic tape. This capability is usually only needed when a new faculty member or a publisher has/sends cards (we have no card equipment on either the HP or the VAX). I will discuss later our plans to provide an alternate means of providing data to academic offices.

About 18 months ago, we were faced with the problem of a saturated system. We were out of capacity, in terms of CPU power and raw resources. In order to resolve this problem we were faced with the need to expand the hardware available. We basically had the options of upgrading to a larger HP3000 or adding another complete system. In meetings with the faculty, there was a concensus reached that breadth and variety were extremely important and that this outweighed the cost of additional staff, etc. After this conclusion was reached, we decided to acquire a VAX 11/750. This decision created another problem regarding terminals. Namely, I did not want to have two groups of terminals each of which was connected to a single system. Our solution was to add a digital switching system between the two CPU's and all the terminals. Normally, a switching system is used when there are more terminals trying to access a system than there are ports. However, this solution allowed us to create a situation whereby any terminal on the network can connect to either the HP or the VAX.

This solution has not been without its own set of problems. First, we have a broad mixture of terminals - HP2640's, HP2621's, ADM5's, VT101's, and others as well as hardcopy terminals and microcomputers attached to the network. The common denominator is that they are all ASCII and use RS 232. That is where any similarity ends. In asynchronous communications, there are two type of devices DCE and DTE (data communications equipment and data terminal equipment). The fun starts when you try to determine exactly how a device acts. Some terminals behave as DTE and some as DCE; some need only three wires in their cable plugs; others need anywhere from four to eight wires, and some need jumpers. (For a thorough discussion on this topic refer to Ross Scroggs's papers in the Proceedings of the last three HPIUG meetings.) In any case, we had to essentially rewire all the terminals so that they could talk to the switching system and then a computer, and also so that the computers could drop the port and thus the switch could free up the terminal. In other words, no terminal is attached permanently to any one system. When dealing with microcomputers, the problem of communications is compounded by the need for software to provide "intelligence". Here again, we were faced with a communications problem compounded by software - the same as for terminals but usually less well documented. The micros themselves are a problem I will discuss later.

Support staff for an operation such as this is always a problem! I do not know of any university that has sufficient staff to support all the activities desired. That this is made even more demanding by multiple systems is perhaps the most obvious statement of the year, if not the decade. It has been my philosophy that generalists are more advantageous then trying to find and afford specialists. I have found it easier to train adaptive people than to broaden the horizons of specialists. In areas where a critical need arises for a short time, those persons can be hired as consultants. Oh yes, we have on several occasions invented the wheel. My staff consists of three people including myself. All of us provide user support to students and to faculty. We operate under the basic philosophy of "we'll help you to do your job". This method of operation has allowed us to provide the range and level of support that we have achieved. For the most part, we have a very satisfied and productive group of users.

This method usually leads to a more educated and self-sufficient faculty than those operations with lots of TA's to do the dirty work. Within this mode of operation, we do very little development work, other than for ourselves and to support our systems. The bulk of our work load is consulting with faculty on problem solving and package selection, and assisting the students with error determination/correction. In addition to the three full time staff, there are about 8 student assistants who provide the majority of the consulting for the other students.

As mentioned previously, we have an HP3000, a VAX 11/750 and a large number of microcomputers. One of the policy decisions made very early was to run "plain vanilla" systems. By this I mean that we will make no changes, of any sort, to an operating system nor will we use an application package that does not include support. We simply do not have the staff or funds to support this type of operation. By choosing "plain vanilla", upgrades and changes to the hardware become relatively simple and usually transparent. In addition to the HP and the VAX, I support a Computer Assisted Instruction Laboratory consisting of 17 Apple II+ micros sharing a Corvus hard disk via an Omni-net system.

Along with these systems, we have had a large number of departments acquire micros for their office which we are expected to support. At many institutions, this leads to chaos. One of the ways we have avoided chaos is that I have budgetary control over any purchase from the academic side of the university that appears to relate to computers. The Computer Center Director has similar control for the administrative side of the University. In order to manage the explosion in micros, we have limited our users choices to those from 3 vendors: IBM, Apple, and Radio Shack. The only permitted exceptions to this are for equipment which will be used exclusively in conjunction with some laboratory equipment. By limiting the choices, we have reduced the number of systems with which we are expected to be familiar. Of course, we still provide only diagnostic and problem solving consultation and system evaluation assistance. At this time we are only supporting (financially) systems which will be for departmental use rather than for the primary use of an individual. All of the systems we acquire have the ability to connect to the Academic Network. Intentionally, we have not provided a link to the Administrative Network. Because of security and other access control problems, we have no plans to attach any micros outside of the Computer Center to the IBM 370. In order to provide data from the Administrative system to the authorized users, we plan to distribute data on IBM PC floppy disks which will be generated by our standard report generators in the various administrative systems. This format will be used as an optional output medium to the standard printed reports.

Supporting two major systems, the HP3000 and the VAX, is not quite as large a task as one might suspect. Both systems are mature and thus require very little hand holding or baby sitting. Since both are also "plain vanilla" and have vendor support our tasks are more preventive than corrective. In this type of operation, however, we are in a constant mode of monitoring the status of both systems in preparation for any eventuality. I feel more comfortable planning than I do trying to recover.

In summary, perhaps I would make two suggestions. First, I strongly believe in the KISS principal. The more simple a system is, the easier it is to maintain and control. Second, regarding micros, complete budgetary control is the only method to know what is going on within an organization.

The "Newcomer Tree": Hands-On Introduction Of Computing Services To The Freshman

Jody Ann Malsbury* and
Jay Martin Anderson

Office of Computing Services
Bryn Mawr College
Bryn Mawr PA 19010
Phone (215) 447-7506

* Present address: Swarthmore College, Swarthmore PA 19081.

In 1978-79, Bryn Mawr proposed the enhancement of "computer literacy" among students and faculty to the program on Comprehensive Assistance to Undergraduate Science Education of the National Science Foundation. One of the seven modules of the proposed program was to provide a hands-on introduction to the use of computing in one's own field of study which could be used freely by freshmen (including incoming graduate students and new faculty). Bryn Mawr received major funding from the NSF for the period 1979-82, and the design and implementation of the "newcomer" program commenced in earnest in 1980.

The newcomer program was aimed primarily at freshmen. Consequently, its design objectives demanded that the subject matter to be dealt with require no more than high school education, for the newcomers should see computing being used in their proposed major subject in a way with which they were already familiar. Secondly, the program should be usable with almost no assistance from the computing center staff. Instructions should be short and simple, and should be contained mostly within the program itself.

As the program evolved, it was clear that the best candidates for preparing examples of the use of computing in the twenty or more academic departments at Bryn Mawr would be students who themselves were majoring or studying in those departments, and who had some computing experience. Consequently, although the first few modules of the newcomer program were prepared by students on the computing center staff, in subsequent years modules for the program have been contributed by students in the elementary FORTRAN courses on campus. It is particularly important to be able to confront the newcomer with the pleasant surprise that she is using the work of her peers, not of software engineers way beyond her in experience.

The curriculum at Bryn Mawr College is structured by divisions and departments. The four divisions (Natural Science, Social Science, Humanities, and Languages & Literature) represent four subject areas in which all students must do a year of work. Within each division are several departments from which the student will choose her "divisional requirements." It seemed reasonable to imitate the structure of the curriculum in the structure of the program that was to introduce the freshman to the use of computing in that curriculum.

Consequently, the newcomer program has a tree-like structure, and is now regularly referred to as the "Newcomer Tree." At the root, the module that runs as soon as the newcomer signs on, the newcomer receives a brief introduction to the computing center at the college. From there, she may move up the Tree to its four branches, representing the four divisions of the college's curriculum. A short paragraph describes some of the usefulness of computing in these subject areas. Above the branches are the stems of the Tree, representing the individual academic departments of the college. At this level, the student may select one or more brief example of the use of computing in that department to investigate.

The root, branches, and stems of the Tree were written by the authors in 1980-81; the authors and members of the computing center staff (advanced students) contributed a few leaves, just so the Tree would not be "bare" in the fall of 1980. By the fall of 1981, many more leaves had been written, including contributions from students in introductory programming courses in 1980-81. Although not every department is represented by a leaf, every division of the college is, and the Tree was ready for formal unveiling to the students who were new on campus in 1981-82.

The leaves of the Tree are programs written in FORTRAN/3000 or Pascal/3000. These standalone programs are the children of the root-branch-stem parent program, using process handling. Leaves are solicited for the Tree when students begin to mature in their programming, usually about the tenth week of a semester's course in computing. The leaf programs must meet these criteria:

- Programs demonstrate the use of computing in a department or program of the college.
- Programs require no more subject-matter

knowledge than could be expected of incoming freshmen.

- Programs are written in FORTRAN or Pascal, run interactively with little CPU time needed and little output generated. If a program produces more than 24 lines of output (one CRT screen), it should be divided into "pages" which the newcomer can control with a simple instruction such as "Press RETURN for more."
- Programs are self-documenting, and instruct the user how to use them. As appropriate, they trap common errors and prompt the newcomer for alternate corrective action.
- We prefer programs that can be run on any terminal; in special cases the characteristics of a particular terminal (CRT graphics, for example) may be used.
- The leaf programs themselves may not use control-Y interrupt. The authors or the computing center staff imbed a control-Y interrupt in each leaf program which causes the leaf to terminate and control to be returned to the stem.
- The student programmer, with the Computing Center's help, must provide a standard "title page" which identifies her program when it is first invoked from the stem.

It is important to note that we have not been strict about programming style or about complete and thorough trapping of errors within the leaf programs. We want the leaves to reflect the personalities and styles of their authors, and we would rather see a leaf program fail unexpectedly than not have that leaf at all. In this way, the newcomer sees both the successes and the failures of her peers, and her expectations are reasonably set.

A major design goal for the "Newcomer Tree" program was that there be no hassles and no documentation for its use. We felt that, to be effective, the Newcomer Tree should be usable on a walkin basis, without the need for elaborate instructions from faculty, staff, or the printed page. The newcomer especially should not need to apply for an "account" at the computing center in order to use the Tree. The most a user needs to be shown, then, is how to turn a terminal on, and how to sign on. This can be conveyed in a few seconds with a student on the computing center staff, or in a half-page instruction sheet available in the computing center, and in the several terminal rooms on campus.

With these instructions, the newcomer turns on a terminal, presses RETURN, sees the colon prompt, and types :HELLO NEWCOMER.COMPUTE and presses RETURN again. From this point on, she is on her own.

A logon user-defined command (UDC) controls the flow of the program. The newcomer sees the same "welcome message" that all users see, and then sees a brief description of the program's purpose. The newcomer is asked to supply her name, and her status (freshman, sophomore, junior, senior, graduate, faculty) on campus. This information is confidential, and whereas names are never used for reporting, some data is accumulated for subsequent analysis.

If the newcomer is a real novice, she may ask for instructions. She will then be told the use of the RETURN key, the BACKSPACE key, CONTROL-Y, and CONTROL-X. As mentioned above, CONTROL-Y is used to abort a leaf program and return to the stem. The student's movement in the Tree is controlled by the same instructions at each stage: "up, down, or quit."

By keeping the instructions required to use the Newcomer Tree to a bare minimum (turn on, press RETURN, type a logon), and the instructions required to move through the Tree equally simple and minimal (up, down, quit, correct typographical errors, and control-Y), we encourage walkin casual use of the Tree. The documentation is short and inexpensive to produce.

The Newcomer Tree has been in regular use for two academic years, 1981-83. The following table shows the extent of use in the two years, plus a one-evening open house in the fall of 1981 which was the formal unveiling of the program.

	1981-82	Open House'81	1982-83
Freshmen	33	30	37
Sophomores	37	1	24
Juniors	52	0	10
Seniors	48	3	9
Graduate	38	16	30
TOTAL	208	50	110

The decline in use of the Newcomer Tree since its formal introduction in 1981 is to be expected; the results in 1982-83 suggest that the Tree may reach about 10% of the student body each year, and, more importantly, perhaps 15-20% of incoming freshmen and new graduate students. Data are also accumulated on the length of each session with the Tree. It is clear that many sessions with the Tree do not get to a leaf at all, yet we judge that ten percent of Tree sessions are over a half-hour long (enough to use more than one leaf, or one leaf many times), and half the sessions use at least one leaf. The average length of a session with the Tree is between twelve and fifteen minutes. Some faculty assigned the Tree to their classes, if only as a controlled exercise in turning on a terminal and signing onto the computer.

A second important result of the Tree has been its impact on the programming classes on campus. Although many students do not wish to be limited in their programming efforts by the criteria we have established for the leaves of the Tree, new leaves are contributed each year. Students who do choose to contribute a leaf are introduced to programming for someone else, and to standards of programming which are not their own: healthy, egoless exercises for budding software artisans. They become more aware of the pitfalls a novice may have in using their work, and begin to taste the meaning of user friendly by placing themselves in next year's freshmen's shoes. Students who write leaves for the Tree learn more advanced features of MPE, and of terminal control.

Finally, the Newcomer Tree has become a model for other hands-on instructional use of the computer in other departments. The Tree is a useful example of a data structure's topic in the last weeks of programming courses, and serves to tie together more theoretical material on trees with a real example.

The HP3000 community may be particularly interested in some of the details of the construction of the Newcomer Tree program, in particular, on terminal control, process handling, UDCs, and JCWs.

In our environment, it is possible to identify all terminals through their logical device number. Whereas not foolproof, a table lookup can tell us (in the root of the Tree) whether the user has an HP CRT, another CRT, a printing terminal, a graphics terminal, etc. Dialup users are treated as if they had printing terminals. Once this has been established, two simple terminal control features can be selected: either the CRT screen clears between steps through the Tree, or several line-feeds are issued between steps through the Tree. Furthermore, it enables us to determine if any special terminal (e.g., graphics) is available if required by a particular leaf.

Process-handling is used in its simplest sense. When the newcomer selects a particular leaf, the root-branch-stem module becomes the parent process. It calls the CREATE and ACTIVATE intrinsics, and launches the leaf as its child process. When the leaf terminates (or when control-Y is pressed, which forces the leaf to terminate), control is returned to the parent, who awakens. In this way, minimal effort is required to add a leaf to a Tree, and the leaf can be independently tested before it is attached to the Tree.

A tightly restricted logon UDC provides access to the Newcomer Tree. The newcomer is isolated from all else on the system, and she is free to leave the Computing Center after typing QUIT while in the root of the Tree. The UDC protects the user from herself while protecting the system from the user.

It happens, of course, that an undetected error occurs in a leaf program. When this happens, a system JCW is set. The user would see the message "terminated in an error state" because of an error in the leaf; she may have forgotten about that error long ago, or she may be concerned that something bad has happened to the computer. Consequently, the root program intercepts the JCW, and, instead of reporting (about the child, or leaf process) that something "terminated in an error state," reminds the user that "some error occurred in a leaf program," and invites her to consult with the staff if she has questions. In this way, the errors of the child are not visited upon its parents.

In summary, the Newcomer Tree has been a successful experience for its primary targets: new individuals on campus. Students are introduced to the potentials for computing in their chosen or prospective field of study in a friendly and unstressed situation. Others on campus have begun to emulate the Newcomer Tree in their own endeavors. Students themselves write and contribute to the Tree. The Tree serves as a useful example of good programming style, the data structure "tree" itself, and cooperative egoless programming.

JOURNAL
OF THE HP 3000 INTERNATIONAL USERS GROUP, INCORPORATED

Computer Based Aids for Curricular Planning, Test Creation, and Processing

Gregory P. Thomas
Peter Wolmut
Multnomah County Education Service District
220 S.E. 102nd
Portland, OR. 97216
Phone (503) 255-1841

Abstract

The Multnomah County Education Service District (MESD) is charged with providing support service to 12 school districts with an approximate enrollment of 88,000 students Kindergarten through 12th grade.

The Measurement and Experimental Research (M & ER) department provides a number of HP3000-based support mechanisms. There are two HP3000 44s and one HP3000 III serving Multnomah. The first system supports the storage and retrieval of course goals and associated test items. Currently this system stores approximately 27,000 goals and 4,600 test items.

The second system provides a means of processing a variety of measurement instruments including achievement tests, criterion referenced tests, check-list instruments, and surveys. Significant features include: interactive test and response sheet description, remote scanning of response sheets, scoring and reporting, updating of student competency records, and creation and maintenance of historic student test records.

Curricular Planning and Test Development

To support curricular planning activities, the MESD supports a computer based system which is currently composed of both goal and test item collections. These collections are organized around the following 13 curricular areas:

- Art
- Biological and Physical Science
- Business Education
- Computer Education
- Health Education
- Home Economics
- Industrial Arts
- Language Arts
- Mathematics
- Music
- Physical Education
- Second Language
- Social Studies

The number of goals varies from collection to collection, ranging from a high of 4475 to a low of 195. The item collections also vary in their depth and breath of coverage. The number of items currently linked to curricular areas ranges from 0 to 3495. The variation in numbers of goals is a reflection of the current curricular thinking about the various areas. The variation in item numbers reflects the level of activity focused on the assessment of certain curricular areas/concepts.

Goal Collections

Over the course of the past decade, the MESD has coordinated the Tri-County Goal project. Master teachers and other educators from the three contiguous counties, Multnomah, Clackamas, and Washington, were charged with the development of a comprehensive goal structure covering all curricular areas. The current collection of 27,000+ goals represents the combined efforts of several hundred professionals in Oregon. The hard copy versions of this collection encompass approximately 4,500 printed pages organized into 13 curricular areas within 16 volumes. This entire collection is maintained by the M&ER department in both hard copy and machine retrievable forms.

Item Collections

At about the same time the goal collections were being assembled, the Northwest Evaluation Association (NWEA) was actively engaged in developing item banks in Reading, Language Usage, and Mathematics. The items which are part of these banks were examined statistically using Rasch techniques.

In addition to this item bank development, the M&ER department has, over the course of years, supported a number of other item generating activities. These activities have served to provide a mechanism focused on client district testing needs. The items generated as a result of this set of activities are examined statistically as they are used in a testing framework. At that point, the items are Rasch analyzed, kept, modified, and/or deleted.

All items which are included are associated with goal(s) contained in one or more of the collections. This "linking" of items to goals allows the user to look for items as an independent activity or as a logical extension of goal development/selection.

System Uses

Educators within the area have utilized the goal and item collections for a variety of purposes. It is possible to use a set of retrieval specifications to focus on grade, curricular area, and key word(s). By making these retrieval specifications broad or narrow, it is possible to "look" at a collection to gain ideas for future educational developments and/or changes. For example, the curriculum as currently defined may focus on a subset of a total set of possible variables. Developing a retrieval statement based on locally developed goals, the collections can be used as a sounding board to test the depth of curriculum coverage. This frequently has suggested additional areas that could be incorporated within a curricular plan.

Once a district has tentatively defined a curricular scope and sequence, the item collections can be addressed to create potential assessment instruments. Items may be retrieved independently of the goal collections or linked to specific goals. In this way teachers and administrators can look at items as "models" of the kind of assessment devices they wish to create. The item collections are not restricted to multiple choice and other paper-and-pencil styles.

This part of the total system is in place as a resource - not as a prescription dictating specific programs of instruction and assessment. By using the goal and item collections, it is possible to quickly search outcomes and items in an entire curricular area(s). It is important to point out that the access to and the use of the system reside in the hands of the independent users. The staff of the M&ER department are at this point in a position to provide technical support and assistance to these users.

Test Processing and Reporting

One of the primary roles the M&ER department plays with client districts is the provision of testing service. Tests which are utilized for assessing student performance are determined by the individual school districts. Therefore, the M&ER department is constantly having to examine its computer capabilities in order to meet the scheduled testing demands. Currently, district wide testing programs operate in both the Fall and Spring of the school year. The number of students processed is quite variable from season to season and year to year. This current school year (82-83), we anticipate processing between 30,000 and 40,000 students. From a processing point of view, a major amount of reprogramming is commonplace whenever a new or different testing program is encountered. Because of budget and staff limitations, a means of quickly adapting to changing demands was needed. The current test processing system was the result of this need. While use of this testing software package has certain limitations and constraints, the M&ER department has found over the course of the last year and one-half that most tests can be accommodated.

The basic system limitations are:

200 items per test
20 goals per test
30 items per goal
5 choices per item
(10 for surveys)

Test/Form descriptions

The starting point in using this system is focused on the data collection form and a description of the actual test. Both descriptions are provided by the user interactively. Data can currently be supplied to the system via Scantron compatible answer sheets, magnetic tape from another computer system, National Computer System scannable documents, and "on system" data files. Using the Scantron data entry mode, the user can enter data from remote locations via the dial-in capabilities all county users currently have. Other means of data entry usually require intervention on the part of M&ER staff. Scantron compatible forms are custom designed and printed within the MESD print shop. Costs vary with the complexity of the form but usually approach $25.00 for producing a plate.

Entering the test description is also an interactive process. The software prompts the user for the various bits of information needed. This includes: number of test items, scoring key, scoring criteria, number of subtests, etc. Entry of a rather complex test description might take an inexperienced user 20 minutes. Once this test description is entered it remains on file within the user's logon until the user changes or deletes the description.

Data Entry

At some point a physical test has to be created. Currently this operation is not computer supported beyond the point of item selection. Modifications are currently underway within the goal/item system to allow a user to select items, format a test, and print the physical test using the already edited copy within the computerized item banks.

Once the test is created, however, the user has the options of scanning sheets which are either NCS (inhouse) or Scantron (remote entry) compatible. Data is also frequently sent to the M&ER department by other local education agencies for processing on magnetic tape. The data entry process establishes data file(s) which are named and used from this point forward; scanning is a one time process.

Scoring/Editing of Tests

Scoring is accomplished by "pointing" the scoring program at the appropriate data file. The test description already provided is consulted in order to accomplish this task. The original test description can be changed and the scoring redone without limit. This allows items to be keyed out at the users discretion.

Multiple marks are often encountered on answer sheets. An interactive edit feature allows the user to edit item responses as well as a number of other fields. Currently the score on both the overall test and subtest cannot be modified by the user. If modifications are made, however, the user is able to rescore the data file to reflect these modifications.

Test Reporting

Three "standard" reports can be produced by the user with a hard copy terminal device. These reports require that the user specify the particular report desired and the data file to be reported. There is no limit to the number of copies which may be produced by the user.

Custom reports are avalaible on an "on demand" basis. The MESD currently supports several generalized report writers which are accessible to the system user. Using one or more of these utility packages, usually it is easy to provide custom reports within short time frames. Several users have also taken the time and energy to learn how to use these report writers and are actively engaged in generating their own reports.

Student Records

The MESD provides the capability for client districts to utilize the computer facility for student record keeping. Approximately 40,000 student records are currently maintained. The testing systems are able to update the individual student competency records, based on test performance. It is also possible to maintain student scores from any or all tests the student takes during the course of his or her enrollment. This system supports movement of the student between county districts as well as withdrawal and re-entry of students. That is, once the student is made a part of the system the record is maintained until the student is matriculated.

Other Capabilities

A number of sophisticated statistical analysis packages are available to the user. These include an interactive Rasch analysis package, SPSS, Report writers, and Graphic reporting systems. The user has the capability of using any or all such packages to examine the data file(s) which have been created during the data entry steps. It is also possible to retrieve historic information and produce reports designed to respond to specific trend types of questions.

This paper was originally presented at: Twenty-First Annual Convention Association for Educational Data Systems (AEDS) May 1983, Portland, Oregon and is reprinted with AEDS permission.

M.I.S. In An Educational Institution As Experienced By Graceland College

David D. De Barthe
Director of Computing Services and Systems
and
Bradley B. Bryant
Systems Analyst
Graceland College
Computer Center
Lamoni, Iowa 50140
Phone (515)784-5167

First Steps

Graceland College first expressed the need for a Management Information System (M.I.S.) in 1974. Various steps toward such a system have been taken over the nine years since that initial start.

In 1977 Graceland changed from a batch oriented little blue machine to an HP3000 Series II and very shortly after that to an HP3000 Series III. A primary emphasis was to get to a real time, user-driven system. The MPE operating system was a major selling point over other similar size machines because it was considered to be simple enough for unsophisticated users. Conversion was primarily a matter of converting the ISAM files to KSAM, recompiling all those RPG programs, and learning to run on a new machine. Some of you may remember the early joys of KSAM.

A computer center staff was established with a part time director and two full time programmer/analysts, each responsible for a different segment of the administrative offices. In the competitive environment that followed, very different approaches to developing systems evolved. In 1978, the following objectives were written which, though generally agreed upon, were implemented differently:

- Graceland's computerized system will be an INTEGRATED information system with departments linked so that when one department changes or adds information to the system, all other users who should have access to that information will get CURRENT information.
- Information system policies and procedures will be written within the framework of Graceland's overall policies and procedures.
- The information systems will be user oriented. The input of data, control of data, and retrieval of all output data will be the responsibility of the user departments.
- Users and Computer Center personnel will co-architect new systems.
- The computer will be used as a tool for those data elements that can be stored, controlled, or retrieved more effectively when computerized.
- Computer systems will allow for interface with other Graceland Information Systems.

Most of the systems development effort, however, was concentrated on designing and redesigning computer based data systems. Little, if any, effort was placed on integrating new and existing Data Processing systems with new and existing manual systems. The lack of effort toward the integration of manual and automated processes hindered the evolution of a comprehensive M.I.S. for the institution. The commission, at that time, appears to have been interpreted "go forth and computerize".

Computer Account Structure

Under the direction of the two programmer/analysts, a computer account structure evolved as shown in Figure 1.

FIGURE 1

ORIGINAL COMPUTER ACCOUNTS

ACCOUNT	GROUPS	
CIS (College Information System)	ADMISSN	(Admissions)
	DATABASE	(general data - only Admissions at present)
	DEVLPMNT	(Development)
	SIS	(Registrar-Student Information System)
	SOURCE	(source programs)
FINAID (Financial Aid)	FINAID	(Financial Aids info)
	PERSON	(Student Payroll)
GAS (Graceland Accounting System)	LEDGER	(Accounting database)
	SOURCE	(source programs)
PERSONEL (Personnel system)	PERSON	(Personnel database)
PROGRAMS (source & object code)	ACPROGRM	(Accounting programs)
	ACSORUCE	(Accounting source code)
	FAPROGRM	(Finaid programs)
	FASOURCE	(Finaid source code)
	etc...	
SYS (system account)	DOCUMENT	(documentation files)
	GRACELND	(locally written utilities)
	LOGFILES	(system logging files)
	PUB	(HP provided utilities)
	UTIL	(externally written utilities)

While the CIS account allowed for easy sharing of data, it did not provide for a convenient way to separate program and documentation files from user work files. Neither did it provide adequate control of user programs and jobs. In an environment where users are encouraged to develop their own jobs, ad hoc reports, and in some cases programs, adequate control must exist if Computing Services personnel are expected to maintain the programs. The first time we got a call from a user office asking us to help "fix" a program and then discovered that it was written in spaghetti structured basic, we saw the need for control. As could be expected, the programmer had graduated and left the area. We also discovered offices writing programs that already existed, either on our system or on contributed library tapes.

The GAS, FINAID, PERSONEL, and PROGRAMS accounts allowed for much greater control by Computing Services staff, but it was almost impossible to share data between offices except by the paper highway or the unsafe ":RELEASE" command.

The GRACELND, UTIL, and PUB groups in the SYS account permitted us to keep track of who was responsible for what but gave users some problems in knowing what group to run what program from. If that sentence confuses you, you'll know why the users complained.

Identification Codes

It is generally understood that one of the foundations of any automated data system is a dependable, unique, and understandable identification system for records. This is true whether you are identifying general ledger accounts, prospective or current students, employees, or anything else. The lack of standardization of identification code became apparent very early in the evolution of our systems, varying from name driven, to social security number, to numbers matched against alphabetized lists, to a very unique combination of the first four characters of person's last name, the first two characters of the first name, and the first two characters of the city where the person lives. Each of these systems, of course, had strengths and weaknesses.

Name systems appear very personal and in small sets have relatively few duplications. However, real problems became apparent when name driven systems were put in IMAGE files. With the tools available now, we are more willing to change an IMAGE key field when someone changes his/her name. It is not an easy task, however, when that field is the key in many different files.

Alphabetized lists quickly got out of order because Graceland's alumni and applicant names did not match the U.S. "normal distributions". How many of you have ever met a DeBarthe? There are over a dozen on Graceland's files and not many more than that in the U.S.

Can you imagine an identification system based on something as dynamic as a person's city of residence? Scratch any consideration of such a system.

This appeared to leave the social security number as the only logical choice, so we tried it. It proved to be almost as bad as the worst of the others. First, most initial contacts with Graceland's publics, although perhaps significant enough to justify some record, did not collect the social security number. In addition, Graceland has a significant population of non-U.S. citizen students who do not have a use for, or access to, a social security number. We found ourselves generating random numbers and then going through the pains of trying to trace down and correct all records when we finally got a valid social security number. With the exception of payroll records, the social security number was not significant anyway.

As a result of the problems with all the identification schemes then in use, we finally decided to go with our own randomly generated identification number for each contact. It could be determined to be unique for each individual and was available upon initial entry to the system. Admittedly, if an operator did not adequately cross check for existence in the system, a person could end up with more than one identification number. This problem is under continual analysis and hopefully is being resolved by forcing a generic search by last name and part of the first name before adding a new record.

Reorganization

We finally recognized that computerization, by itself, does not improve the quality of the data or the information provided. A total, integrated approach to management AND system design was needed in order to best use our information sharing capabilities.

Changes came about slowly. However, over a period of years, the part-time director and one of the systems analysts left for greener pastures (within a month of each other). The computer center became Computing Services and Systems and was reorganized with a full time director, a systems analyst, and a programmer. The programmer was later replaced with student programmers, and a secretary was employed. Since then, a full time programmer/analyst has been added. A team approach has been implemented and progress toward an integrated user-driven system has increased.

One of the first steps toward integration was to get to a consistent, integrated computer account structure. Borrowing from the best of former approaches, the structure shown in Figure 2 was developed.

FIGURE 2

INTEGRATED COMPUTER ACCOUNTS

```
ACCOUNT    GROUP

MANUALS               Contains all documentation for user manuals
                      as many offices as needed.
           AUT        Automatic Jobs
           PUB        Work Group
           UDC        User Defined Commands
           UTL        Utility Processes
           TOOLS      Programming Aids
           TRAIN      Training Manual Materials

PROCESS               Contains all processes except for Hewlett-
                      Packard supported.
           Identical to the MANUALS account except for the
           omission of the TRAIN and AUT groups.

GIS                   Graceland Integrated System
                      (*** represents the 3-character office
                      code.)

           ***WORK    General work group for ***office
           ***DOCU    Contains general documentation for
                      ***office
           AUTOJOB    Contains automatically streamed jobs
           DATABASE   Contains all of Graceland's integrated
                      databases
           PUB        Temporary holding place for transient files
                      plus an SL

TEST                  Account for creating and testing processes.

           ***SOURC   Contains implemented sources (PROTECTED)
           ***WORK    General test work group for ***office
           DATABASE   Contains all of the integrated test
                      databases for Graceland's offices
           PUB        Temporary holding place for transient
                      files, plus an SL
           PROCESS    Contains processes being created & tested
           SOURCE     Contains sources being created & tested
           MANUALS    Contains documentation for processes being
                      created & tested

SYS                   Hewlett-Packard-supplied processes and
                      supporting files

           PUB        Contains HP supplied processes
           SYSOPER    Home group for System Operator (emergency
                      backup files)
           AUTOJOBS   Contains automatically streamed jobs for
                      the system
           >>> Other HP created groups <<<
```

Within this revised ACCOUNT structure, there are four main ACCOUNTS that comprise the administrative M.I.S. structure. The MANUALS, and PROCESS accounts are accessible by all users but may be written to by computing services staff only. The Computing Services Program Librarian (secretary) is primarily responsible for the control of changes made to files in these accounts. The Program Librarian is also the only user in the TEST ACCOUNT who can create or replace files in the ★★★SOURC groups. System and GIS account manager capability is given Computing Services personnel only.

The TEST ACCOUNT is structurally similar to the GIS ACCOUNT. All groups in both of these ACCOUNTs, except PUB and DATABASE, have the FILES disc space limit imposed. Each administrative user has a unique USER-NAME assigned, and logs into the ***WORK group corresponding with their office. By assigning each user to the office WORK group, and placing GROUP passwords on all non-WORK groups, we have greatly minimized the problems caused by migrating "junk files".

Disc Space

We were always out of disc space, and have been for about two years. As most of you know, the more you add, the more you gobble up. We finally gained control of the disc space problem by judicious use of the FILES option on accounts and groups. About two years ago, we assigned all the disc space the system had (460 meg), reserving calculated amounts for spool files and virtual memory, so that each office had an appropriate share. It then became the responsibility of each office to keep their "junk" cleaned up to the point that they had usable work space. If they must have more space, they find it in their own groups or get some other office to give them some. This, of course, does not solve the problem of allowing for additional applications and growth. It has, however, stopped run away inflation of disc storage. When we add more discs to our system, a percentage of the space will be allocated to future expansion.

To help users maintain control over the files in their work groups, an automated system has been implemented which will purge files which have not been modified within the last month. These files are recoverable from Sysdump tapes.

Programming

Graceland Computer Science and Computer Engineering students have been in high demand by industry. We decided to try them ourselves and have had excellent results. There is always a brief period of training and orientation but students are already conditioned to learn. We generally hire juniors and seniors who have completed their programming and information systems courses. There is intense competition among students for the available positions, and without exception they have done excellent work. If you try student programmers, be sure you use walk-throughs on a frequent basis and reviews when a program nears completion. In Graceland's current operation, the user office is responsible for the system specifications and terminal resources. Computing Services and Systems is responsible for programming supervision and training.

Future Steps

In 1982, Dr. Joe Hanna, President of Graceland, gave "respectability" to M.I.S. by instituting a long range plan calling for a comprehensive M.I.S. This began a struggle to define and come to grips with what M.I.S. should be for Graceland in the 80's and beyond. The first step was to have an M.I.S. Steering Committee appointed, consisting of two representatives from each of the four Vice-Presidential areas. After months of data flow diagramming, reading, and planning, a set of new objectives has been written and is in the process of being approved. We have generally concluded, however, that M.I.S. is whatever the institution needs for it to be, so do not ask someone else to define one for you.

While the analysis and definition of M.I.S. has been going on, we have proceeded with the design and programming of a common address system for the institution. If you are lucky enough to have all your offices sharing an address file instead of each office maintaining their own, we envy you. We have six different automated systems that we are aware of and we have not counted the office manual files. Implementing one address system means all present systems must be redesigned to use the central system. Our approach has been highly influenced by the article, *A True Data Base Environment*, Computer World, February 7,1983, written by Dave Ackley. Because of the ease of writing update programs with TRANSACT/ 3000, we are suggesting to our offices that they design their subject data base, enter the definitions into DICTIONARY/3000, and then write update programs for the new systems. In order to keep existing information systems operational and still develop the new systems, programs are being written to 'back load' data to the old data bases and KSAM files.

Conclusion

M.I.S. will never be a final product. If it should be killed or die a natural death, it will be reincarnated under some other name. At Graceland, we are confident that users will become more and more involved with controlling their own information destinies. Whether this is by more user friendly languages or more user friendly machines or some other approach does not really matter. They are getting involved at an increasing rate, and the trend will continue. We are also confident that to meet the long term needs of any institution, information must be shared by various institutional offices and officers. Under any name, M.I.S. is just that - sharing information under the control of the people who have need for it. It's going to be fun trying to get there.

Freedom To Choose Or KSAM An Alternative To IMAGE

Darrell W. Conger
Amarillo College
Computer Center
P.O. BOX 447
Amarillo, Texas 79178
Phone (806) 376-5111 x 2075

This is a paper on converting from Univac to Hewlett-Packard with emphasis on standards, procedures, and software used during and after the conversion.

Background

Amarillo College is a community college located in the Texas Panhandle with an enrollment approaching 6000 during the fall and spring semesters. The computer usage history has been a progression through an IBM 1401, a Univac 9300, a Univac 9480 and finally to Hewlett-Packard in 1979. During the last year of the Univac contract, a decision was made to open the computer hardware and software bid process to all vendors rather than keep renewing the Univac contract. Eight vendors including Univac, IBM, NCR, Burroughs, Honeywell, Prime, DEC, and Hewlett-Packard (HP) bid. HP received the contract with a bid of two HP3000 III's, one for academic and one for administrative processing.

Conversion

Conversion is a real concern when changing vendors. The fear of conversion and not knowing how long it will take or even if it will succeed, keeps organizations from considering changing computer equipment. Management's failure to develop and promote the necessary standards and procedures to at least provide an opportunity to opt for the equivalent performance and reliability at a lower cost, is tragic.

Work

Amarillo College had accumulated 600 COBOL and 120 Assembler programs during the 10 years with Univac. This made the conversion a formidable one. The process was made acceptable by having set program development standards and operating procedures in the Univac era. COBOL, a high level language, was set as our administrative development tool. The Univac COBOL and Assembler programs were used for "batch" processing and terminal entry respectively.

Database

One of the important considerations during the conversion was how to convert the database. The two logical choices were IMAGE, HP's data base management system, and the Keyed Sequential Access Method (KSAM), a method of accessing files indexed by keys. The decision was made for a two step approach. First, convert to KSAM and then, when time permitted, convert to IMAGE. (The time needed to learn IMAGE was considered a liability at this critical stage). Having used Univac's Indexed Sequential Access Method (ISAM) gave the green light for the KSAM choice.

Differences

The differences in the Environment and Data Divisions of the Univac and HP COBOL were significant. It should be noted that KSAM files in HP COBOL did not have the standard ANSI ISAM coding structures. (COBOL II does have these standard ANSI coding entries.) This lack of standard coding caused a potential increase in time for program changes. These differences led to the development of the COBOL subroutine to handle all KSAM I-O for the college's database.

KSAMIO

The Univac terminal entry assembler programs used an in-house common ISAM file handler routine for accessing the database. This logic structure was used in developing the KSAM file handler. It was structured in such a way that adding a new KSAM file required changing 2 lines of code and also adding 6 lines for a table entry. Depending upon the type of KSAM file there could be one or two additional logic entries.

Calling Programs

COBOL programs require a "copy" clause for the KSAM table with additional "copy" clauses only for the files being accessed in the Working-Storage Section. To access any file, one issues a "call" to KSAMIO supplying it with the file, function, and key (if needed). The handler returns one of two things, either the desired record or an "error" code. Share and Lock capabilities are used to protect multiple accesses. Currently, our handler has 32 different KSAM files while we may have actually 50 physical files due to using file equates. It is a dynamic and flexible "common" way to access our integrated database.

Current Status

We have now used this I/O handler for 4 years and do not intend to take the second conversion step to IMAGE because of the following reasons:

1. Program development has been simplified.
2. It is easy to understand the file organization.
3. File I-O has been standardized.
4. Adding new files is a "snap".
5. It facilitates prompt program maintenance.
6. File recovery requires no more than 15 minutes.
7. Conversion to COBOL II let us continue our approach.
8. Advantages of IMAGE versus KSAM are not compelling enough to change.
9. A KSAM database can be "covered" with Quasar's QUIZ report writer generator for "MIS" and "DS" reporting.

Technical Description

KSAMIO is written in COBOL II. The table of files is similar to the one described in the HP KSAMUTIL (COBOL section) manual. Two restrictions to the file handler are: (1) a key must begin in the first position of the record and, (2) only one key is supported per record (no multiple keyed records). The file table contains the following on each KSAM file: (1) file number; (2) file name; (3) record size; (4) key length; (5) previous operation completion code; and (6) open close status code. The main program calls the routine and furnishes the following parameters: (1) file number; (2) I/O function desired; (3) key (if used in I/O routine); and (4) pointer read type if needed. The subroutine passes the record accessed to the calling program if the I/O was successful. Completion/error codes for I/O are always passed back to main program. First time entry into I/O routine opens file and sets "open" status for future accesses. All KSAM files are set for input/output with dynamic access. Files are shared and locked until pointer procedures are completed.

Summary

We feel that COBOL, KSAMIO, SORT, and QUIZ have given us the tools needed to satisfy most any service request. Our backlog of application requests has been reduced from 2 years to 3 months. The programmer/analysts spend more time on new development than on program maintenance because changes no longer take the biggest percentage of our working hours. We feel very much at ease in this environment. You might say that if it is not broken, we leave it alone.

Santa Rosa Student Records System

Bob Schooling
Santa Rosa Junior College
Director of Computing Services
1501 Mendocino Avenue
Santa Rosa, California 95401
Phone (707) 527-4385

The *New Student Records* (NSR) system was implemented at Santa Rosa Junior College (SRJC) in Spring of 1980. Since that time, in excess of 20 two-year and four-year colleges have adopted the system. The California community colleges have formed the California Community College Software Consortium (CCCSC) in order to promote the sharing of administrative accomplishments in computing.

The system is interactive with users controlling the input of information in all aspects of the system. This is accomplished via terminals located in the user offices.

Functions of the System

The current version of the system provides for the following functions to be accomplished via terminal:

- Course Catalog
- Schedule of Classes
- Registration/including fee calculation (automatic)
- Change of program
- Various lookups
- Student accounts (debts to the college—interacts with registration)
- Various course displays including rosters
- Positive attendance maintenance
- Maintenance of all student data including applications
- Maintenance of facilities information
- Graduation information
- Financial Aid
- Vocational Education Data (VEDS) Reporting System
- Work Experience
- Test Scores
- Academic History

A Little Detail

There are several key functions of a student system that everyone inquires about—building the schedule of classes, registration/change of program and transcripts. Consequently, short descriptions of these modules are in order.

Building the Schedule of Classes

There are two key sets of information involved. One is the schedule itself; the second is the Master Course information which represents all curriculum approved classes.

As a starting point, the schedule from a year prior to the desired schedule is "rolled" forward (e.g. Fall 83 to Fall 84). The department chairs use this "rolled" schedule as their starting point. They indicate more or less sections, changes to days and times, etc.

The schedulers enter all of these changes into the system via terminal. The program edits the information and checks for overlapping uses of facilities (it is maintaining room utilization information). If a course is being offered that was not present in the "rolled" schedule, then the Master Course information is referenced. This reference causes the screen to be "painted" with all of the relevant, nontransient data. The scheduler then adds dates, times, instructors and other term oriented information.

This process continues as long as necessary. When it is time to produce the printer copy, the section numbers are frozen and we move towards registration. Past this point, new courses or sections can still be added and changes can still be made. Changes can be made very rapidly; an example would be opening new sections in response to demand for a course.

Registration

All registration is done via terminals with integral printers. A schedule/receipt is generated immediately. The program checks for: Delinquent debts (issues a warning prior to registration); Time conflicts; Class limit (including concurrent sections)

All fee calculations are automatic, and fees are itemized on the schedule/receipt. It includes nonresident, material, health, use, parking and associated student body fees. Under specific guidelines, fees can be waived, deferred or partially paid.

The terminal is functioning as a cash register. When an operator "rings off", the terminal will print out a form for the cash box that shows fees collected, waived and deferred by type (e.g. material, health). This printout is then used to reconcile the cash.

The registration process is menu driven. Other items on the menu include:

- Lookup of student data
- Cancellation of a registration
- Class roster (alphabetized list of all students in a class)

Summary data on a section (e.g. title, limit, days, time, current enrollment)
Lookup by last name
Entry of grades (used only for late grades)
Change of program
Drop by section

Change of Program

The system generates a new schedule/receipt any time a change of program is made. Change of program can be intermixed with registration. The system keeps track of all fees. For example, if a student originally paid $20.00 in material fees as follows: BDP52 $10.00 and BDP61 $10.00

The student then drops BDP61 and adds a course with a $15.00 material fee. He/she would be charged $5.00. If the student has a refund coming, the amount is tracked and either applied to a future transaction or detailed in a refund report.

Academic History

The most recently completed module—April 1983—is Academic History. The history module incorporates information necessary to produce transcripts. The transcripts are generated, on demand, on a printer in the registrar's office.

The historical information is updated programmatically at the end of each term. In addition, updates, such as grade changes, are made by the registrar's staff.

Other information such as financial aid, certificates awarded and various elements of indicative data also are maintained. The history module is a boon for counseling, research, course repetition checks and evaluation of degree, certificates and general education requirements.

Summary

The NSR has been an excellent tool for SRJC. Its reception by other colleges has far exceeded our expectations. The resultant consortium has permitted us to share software and expertise as well as leverage significant discounts from vendors for hardware and software.

We have completed our work in a transactional sense. That is, all of the standard, workaday tasks have been addressed. We're now ready for the informational phase.

The informational phase will be the harnessing of the power of the data in the data bases. This harnessing will be through models, special reports and other presentations of the data to management.

We have begun this process of management involvement by placing micros in various managers' offices. The managers are being trained on microcomputer applications (spreadsheet graphics, database, etc.) as well as use of the micro as a terminal to the HP3000. By the end of 1983, we plan to have our managers accessing student data, financial data, a course relationship model and various other information. This system participation by management should strengthen the bond of the student system to the institution.

Technical Information

- All system users have one model or another of the Hewlett-Packard 3000 computer.
- All programs are written in ANSI COBOL 74 using structured programming standards.
- Hewlett-Packard IMAGE data base is used throughout the system.
- Some use is made of V/3000—a screen formatter—requiring some terminals with block mode.
- Substantial use is made of QUERY for reporting. QUERY is a companion product to IMAGE.
- Security is provided through passwords relevant to the basic accounting structure plus database passwords plus read/write lists in the database.
- Information such as fees and dates are not hardcoded in the system, but rather, are accessed from a one-entry data set making them easy to change.
- Considerable use is made of menus and subprograms.
- SRJC, with its 24,000 students, has a total of 25 terminals whose primary use is for some student record related function.

Anyone wishing further information should contact: Bob Schooling at the address listed earlier.

High School Computer Literacy Support by Hewlett-Packard

Gary Gubitz
Hewlett-Packard Co.
Corporate Training, Building 18M
1819 Page Mill Road
Palo Alto, California 94304
Phone 415 857-4167

Hewlett-Packard is involved in a variety of ways to increase computer literacy at the High School level. This article describes model partnership programs within the State of California. The emphasis of each of these programs varies relative to the target audience and the instructional objectives to be achieved.

Peninsula Academics Program This program is a local business, community, and education partnership for 10th–12th graders who are potential dropouts. The objective is to reduce the high school dropout rate, to reduce the youth and minority unemployment rate, and to reduce the number of unfilled entry level jobs in local high-tech companies. By providing a package of resources from industry (computers and electronics equipment, loaned instructors, volunteer industry mentors and summer jobs) this partnership program has had tremendous results. Ten HP85 computers plus peripherals were donated to this program. After its second full year of operation, student proficiency, attendance, self-esteem, grades, and self-motivation have all increased well beyond our expectations. Currently we are working with the California State Department of Education to provide a mechanism for replicating this program extensively throughout the state.

The Institute of Computer Technology (ICT) The Institute is a public school which provides innovative programs to students in kindergarten through 12th grade and to adults. The Institute is a creative response to the problem of educating students to meet the growing needs of Silicon Valley industries for skilled professional-technical employees. ICT offers:

- Accelerated courses for college-bound high school students planning careers in computers and related fields;
- State-of-the-Art training for students and community members seeking entry-level technical positions in the high tech industry;
- Staff development for educators in using microcomputers for instruction and school management;
- A forum for industry and education to join in a dynamic new partnership to meet the needs of students, the community, and industry.

ICT is a joint venture between industry and education. Industry managers and technical personnel play an active role in the development of the Institute's curriculum. The first classes were held on January 3, 1983. Ten HP86 Personal Computers plus associated peripherals were donated to this program.

HP/High School G.I.F.T. Program (Grants for Instruction in Future Technologies): The pilot program has two broad objectives:

- Increase general student awareness and competence with personal computers;
- Develop in high school students the fundamental skills necessary to succeed in the dynamic high-tech world of tomorrow. Skills such as problem-solving, systems analysis, critical thinking, learning strategies, and decision-making are important to the long-term success of our young people.

To achieve these objectives, HP has provided a package of resources to each of 14 high schools in California. The schools were selected on the basis of their proximity to an HP facility, their having a significant number of minority students, or an existing computer curriculum. The package of resources each school received was ten (10) HP86 Personal Computers, an HP86 2-day Training Course, and a volunteer HP Liaison, a technical advisor to work personally with each school to answer any questions that may arise. Also, HP is, 1) providing a forum for these teachers to share curriculum and staff development ideas, and 2) having expert teachers actually design two courses in computer literacy and programming that will be shared among all the schools. Initial feedback from these schools indicates that HP liaisons are the key to successful incorporation of the computers in the high school curriculum. We recognize that other companies are involved in computer contribution programs, and we feel the ones that will make a significant impact will include teacher training and personal assistance along with the computers and software.

A Home Economics Dietary Analysis System

Beth Craig
University of Tennessee at Chattanooga
Academic Computing Services
Chattanooga, Tennessee 37402
Phone (615) 755-4389

Every bookstore is full of books on diets while the libraries report these are high on their lending lists. With such a great interest in this area, it is not surprising that computers are being enlisted to help us select food items with a good nutritional value. A system of computer programs has been in use for doing just this at the University of Tennessee at Chattanooga.

The Dietary Analysis System we use consists of five COBOL programs. The system was authored some years ago by a nutrition student named Maria Fabrega, who was a student at Clarke College, Dubuque, Iowa. Sister Mary Kenneth Keller, Head of Computing Services at that college sent us listings of the programs. (Maria returned to South America after completing her studies; according to Sister Mary Kenneth Keller she "is now a very successful dietician and has become well-known as a dietary expert in several South American countries.")

The system developed at Clarke College had five programs. Three could be termed as "for one-time use" since they were run only to check and enter the two data sets required. (We had to enter the data manually but could then use the programs to check for accuracy). UTC chose to write the data to KSAM files, though EDITOR (ASCII) files would have been equally suitable.

The two remaining programs are used by the students to enter the data (editing program) and to generate a dietary analysis report sheet. The report lists all the food items, their corresponding nutrients, the appropriate Recommended Dietary Allowances (RDA) and the necessary (or superfluous) amounts of nutrients provided by the diet. In both programs, the numbers used to designate the food items are based on the USDA Handbook #8. In addition to specifying the RDA number for each food item, the student must also enter the approximate number of grams consumed, to the nearest whole number.

The system is designed to analyze either a general or a therapeutic diet for an individual. The therapeutic diets available are:

Pregnant
Lactating
Low-calorie-A (1000 calories)
Low-calorie-B (1200 calories)
Low-calorie-C (1500 calories)
Low-protein-A (30 grams)
Low-protein-B (40 grams)
Low-sodium-A (5000 milligrams)
Low-sodium-B (10000 milligrams)
Low-calcium
Low-potassium

In addition to categorizing the client by type of diet, the student indicates other variables such as child/adult, male/female. Age, height and weight also are entered in order to produce an effective picture of the person's diet in relation to a peer group.

The Dietary Analysis system relies on two data files. The first contains a record for each one of the 2848 food items in the RDA Handbook with the amounts of 15 common nutrients in these foods. The second data set contains information based on the Department of Agriculture's recommendations, regarding minimum food requirements by sex and age. Two years ago, the second data set was updated at UTC to reflect recent changes in RDA daily requirements. We understand that a new RDA Handbook, which contains RDA numbers for fast food items, is available but so far we have not revised the appropriate data set.

The printout (Figure 3) illustrates the type of information that the students or dieticians have access to once the appropriate client information has been processed. In addition to the fifteen food nutrients, the caloric intake is shown as well as how well the client's diet meets the daily RDA requirements for each item.

When we received the programs they were designed to run in batch format with the user having to enter the information through punched IBM cards. We discovered that most Home Economics majors were not familiar with the computer and that they found having to code the cards and then key-punch them for the editing program was tedious and difficult. The students also encountered trouble in competing with Computer Science and Engineering majors who needed the key punches for their extensive programming activities.

In 1979 UTC started to switch its students from the card/key punch to the terminal mode of entry. We decided that the Home Economics system could be changed over too, and that this would achieve better results. All we needed was an interactive program to the diets as they were entered by the students.

Since the original programs were written in COBOL, it seemed sensible to stay with that language for the interactive program. However, a test program rapidly revealed a problem - COBOL was not well suited to interactive use! As soon as the last digit of any answer was entered but before the RETURN key could be pressed, the program moved to the next question at a speed that was disconcerting for the user. While an experienced programmer or user would learn to live with this, we doubted that a one time user, with little or no computer knowledge, would find this approach acceptable.

Luckily, this problem was solved by Jeff Kell who works in UTC's Computer Center. He has written a series of SPL subroutines which can be called from a COBOL program. The routines overcome the previous problem and also allow the programmer to indicate a range into which the answer might fall. For example, if the acceptable codes for a certain field are 1, 2 and 3 then the user is told that he/she has put in a wrong answer when anything other than those three codes are entered. If the user enters more characters than are designated in the program, a warning message is generated. Because the SPL code provides these automatic warnings, the programmer is relieved of having to include these messages in the program. The SPL routines also check to see if the user has aborted the job; if so, the file is closed properly. Last, but not least, the user can take as long as he wants to answer the interactive questions, thus overcoming "computer anxiety".

The interactive program was relatively easy to write thanks to the SPL routines. As the user enters each field via the terminal, the program checks the field for accuracy; at each stage provision is made for the user to make corrections. Once the programmer has set up a pattern to check one field, the pattern can be repeated for other fields, using the HOLD command. Thus, only a few statements need to be changed each time a new field is checked. For example, here are the statements that check the client's weight:

```
360-CHECK-WEIGHT.
    MOVE "                    " TO OUT--INFO.
    PERFORM 800-TIO.
    MOVE "  ENTER CLIENT'S WEIGHT TO NEAREST POUND"
         TO OUT--INFO.
    PERFORM 800-TIO.
    MOVE "  (DO NOT USE DECIMALS OR FRACTIONS > "
         TO OUT--INFO.
    PERFORM 800-TIO.
    MOVE "F?ENTER WEIGHT > ZZ9R(1/350)" TO OUT--INFO.
    PERFORM 800-TIO.
    IF USER--ABORT
         GO TO 820-PROGRAM-ABORTED.
    MOVE IN-NUM-3 TO WEIGHT.
    MOVE " ?           CORRECT WEIGHT?   Y OR N"
         TO OUT--INFO.
    MOVE IN-NUM-3 TO OUT-NUM-3.
    PERFORM 800-TIO.
    IF USER-RESPONSE = "Y"
         NEXT SENTENCE ELSE
              GO TO 360-CHECK-WEIGHT.

370-CHECK-HEIGHT.
```

A few words of explanation - each time a question is to be printed on the screen or printer terminal, the SPL routine is called by the statement PERFORM 800-TIO. The answers received from the user are stored in fields named IN-NUM-2, IN-NUM-3 etc. where the number refers to the actual number of digits in the answer. The same naming sequence is used for OUT-NUM, which as might be expected, displays information to the user.

With these comments in mind, note that it would be relatively easy to copy the code for weight checking, then make minor changes to apply it to program areas for checking height, age etc.

The interactive program feeds information entered by the user into a COBOL (ASCII file) which contains 80 byte records. Although the record length is fixed, there are two types of records. The first type, which has a "1" in the first column, contains information about the client such as height, weight, age, name, sex and diet type. The second type of record, designated by a "2" in the first column, allows three food items per record. As the user enters information it is written to the disc file by the program. When the user has finished entering data, a UDC routes the data in the file, links it to the data sets and releases the results to the printer.

The program has been used at UTC for over three years with the Home Economics and Health, Physical Education and Recreation Departments being the prime users. (Since it does require a passing knowledge of the RDA Handbook, only students who have access to it have used the system so far). In addition, dieticians from the Erlanger Hospital, several local doctors and nutrition specialists from the Tennessee Valley Authority have used the program.

The procedure for using the program is described in Figure 1 an actual run of the program is shown in Figure 2. Figure 3 is a copy of a listing showing the data generated for the students.

When the conversion to an interactive program had been completed, we wrote about it to Sister Mary Kenneth Keller and at the same time asked how she would respond to other educational institutions who want to use the program. In her reply she stated, "We would be happy to have you provide copies of the dietary analysis programs . . . to any non-profit organization."

The dietary system has been used at UTC for over four years and the students appear to enjoy using it. Instructors usually have their students list the items that each ate in a three day period and then analyse them. The students are often surprised at the results!

Figure 1 - Procedure for Using Interactive Editing Program

1. The user logs onto the HP3000 system and indicates with UDC-linked EDIT command the name of the file to which the data is to be sent (This file would have been created previously with the BUILD command). The user is given the chance to make sure he/she is using the right program and to exit if so desired.

2. The user is prompted for the client's name which can be entered in any format, provided that no more than 20 characters are used.

3. The user then works through a series of questions about the client's type, age, weight, height and type of diet. As each answer is entered, it is redisplayed to the user with the opportunity to make changes.

4. The user is then asked how many food items are to be entered as this is needed to provide a loop for entering information on the food items.

5. Depending on the number of food items, the user is taken through a loop that asks for the RDA number, name of the food and number of grams consumed for each item.

6. After the loop for food items has been completed, the user is given the chance to (1) enter data on the same client (this is useful if the diet covers several days with each day to be analyzed separately), (2) start entering information on a new client or (3) sign off.

Figure 2 -RUN OF DIETARY ANALYSIS PROGRAM

```
:hello student.chshomec
ENTER ACCOUNT PASSWORD:

ENTER USER PASSWORD:

HP3000 / MPE IV C.KO.A3 THU. Feb 8, 1982, 11:12 AM

:Edit  filename


1 - HP terminal
2 - SOROC terminal
3 - INFORTON terminal

0 - Any other terminal


Enter terminal type  (0-3)>0

WELCOME TO THE DIETARY ANALYSIS PROGRAM

DO YOU WANT TO ENTER DIET DATA?
 ANSWER Y FOR YES OR N FOR NO>Y

YOU WILL BE ASKED QUESTIONS ABOUT YOUR CLIENT
(E.G. NAME, AGE, TYPE OF DIET).  PLEASE USE
UPPER CASE LETTERS TO RESPOND TO QUESTIONS THAT
REQUIRE ALPHABETIC ANSWERS.

ENTER CLIENT'S FIRST AND LAST NAME> JOHN SMITH
JOHN SMITH                 CORRECT? ANSWER Y OR N>Y

WHEN ASKED FOR CLIENT TYPE ENTER EITHER:
         M FOR MALE
OR       F FOR FEMALE
OR       C FOR CHILD

ENTER CLIENT TYPE>M
M        CORRECT CLIENT TYPE? Y OR N>Y

PLEASE ENTER CLIENT'S AGE TO NEAREST YEAR
(DO NOT USE DECIMALS OR FRACTIONS)
IF AGE GREATER 50 ENTER 51

ENTER CLIENT'S AGE>26
26        CORRECT AGE?  Y OR N>Y

ENTER CLIENT'S WEIGHT TO NEAREST POUND
(DO NOT USE DECIMALS OR FRACTIONS)
ENTER WEIGHT>178
178       CORRECT WEIGHT?        Y OR N > Y

ENTER CLIENT'S HEIGHT TO NEAREST INCH
(DO NOT USE DECIMALS OR FRACTIONS)
ENTER HEIGHT> 65
65        CORRECT HEIGHT?    Y OR N > Y

ENTER TYPE OF DIET (GENERAL, LOW-FAT ETC.)>GENERAL
GENERAL           CORRECT DIET?  Y OR N > Y

ENTER NUMBER OF FOOD ITEMS FOR CLIENT > 2
002    CORRECT NUMBER OF ITEMS?       Y OR N > Y

NEXT ENTER AN RDA # BETWEEN 1 AND 2483
ENTER RDA NUMBER > 1317
1317        CORRECT RDA #?     Y OR N > Y

ENTER FOOD NAME (MAXIMUM 12 CHARACTERS)   MARGARINE
MARGARINE              CORRECT FOOD NAME?  Y OR N> Y

ENTER FOOD QUANTITY (TO THE NEAREST GRAM)
(DO NOT USE DECIMALS OR FRACTIONS)

ENTER QUANTITY> 50
0050         CORRECT QUANTITY?     Y OR N> Y

NEXT ENTER AN RDA # BETWEEN 1 AND 2483
ENTER RDA NUMBER > 977
0977        CORRECT RDA #?   Y OR N > Y

ENTER FOOD NAME (MAXIMUM 12 CHARACTERS) > EGG
EGG                    CORRECT FOOD NAME?   Y OR NO > Y

ENTER FOOD QUANTITY (TO THE NEAREST GRAM)
(DO NOT USE DECIMALS OR FRACTIONS).

ENTER QUANTITY > 110
0110         CORRECT QUANTITY?     Y OR NO > Y

DO YOU WANT TO:

        1.  ADD MORE DATA ON PREVIOUS CLIENT
        2.  ENTER DATA ON NEW CLIENT
        3.  STOP ENTERING DATA?

ENTER EITHER 1, 2 OR 3 > 3
PLEASE CONFIRM YOU WANT TO STOP ENTERING DATA
DO YOU WANT TO STOP?  Y OR NO > Y

YOUR DATA HAS BEEN ENTERED ON THE FILE:
ANOTHER PROGRAM WILL PROCESS YOUR DATA

END OF PROGRAM
```

NAME MARY SMITH
AGE 51
DIET GENERAL
WEIGHT 163LBS
HEIGHT 66IN.

DIETARY ANALYSIS

FOOD NAME	AMT	ENERG KCAL	PROT GM.	FAT GM.	CHO GM.	FIBER GM.	CAL MG.	PHOS MG.	FE MG.	NA MG.	POTAS MG.	VIT.A I.U.	VIT.B1 MG.	RBFLAVN MG.	NIACN MG.	VIT.C MG
MARGARINE	0050	360	0.3	40.5	0.2	0.0	10	8	0.0	493	11	1650	0.00	0.00	0.0	0
EGG	0064	110	7.1	8.2	1.5	0.0	51	120	1.0	164	93	691	0.05	0.17	0.0	0
BREAD	0048	371	2.0	40.2	0.0	0.0	0	16	0.2	6	0	81	0.00	0.01	0.4	0
JELLY	0028	76	0.0	0.0	19.7	0.0	5	1	0.4	4	21	2	0.00	0.00	0.0	1
BANANA	0175	148	1.9	0.3	38.8	0.9	14	45	1.2	1	647	332	0.06	0.10	1.2	17
ORANGE	0204	95	1.4	0.4	24.4	1.0	87	34	0.4	2	420	408	0.20	0.08	0.8	91
MILK	0448	264	18.8	8.9	26.8	0.0	640	501	0.4	273	764	358	0.17	0.94	0.4	4
PEANUTS	0075	438	19.5	37.3	14.1	1.8	55	300	1.5	313	505	0	0.24	0.09	12.9	0
COFFEE	0448	4	0.0	0.0	0.0	0.0	9	17	0.4	4	161	0	0.00	0.00	1.3	0
KRAUT	0120	21	1.2	0.2	4.8	0.8	43	21	0.6	536	168	60	0.03	0.04	0.2	16
HAMBURG	0090	257	21.7	18.2	0.0	0.0	9	174	2.8	42	405	36	0.08	0.18	4.8	0
SPAG MEAT SA	0336	450	25.2	15.7	52.4	1.0	168	319	5.0	1367	900	2150	0.33	0.40	5.3	30
SPINACH	0102	23	3.0	0.3	3.7	0.8	115	44	2.1	53	339	8058	0.07	0.15	0.4	19
APPLE	0185	107	0.3	1.1	26.8	1.8	12	18	0.5	1	203	166	0.05	0.03	0.1	7
MARGARIN	0050	360	0.3	40.5	0.2	0.0	10	8	0.0	493	11	1650	0.00	0.00	0.0	0
COFFEE	0448	4	0.0	0.0	0.0	0.0	8	17	0.4	4	161	0	0.00	0.00	1.3	0
BREAD	0056	433	2.4	46.9	0.0	0.0	1	19	0.3	0	0	95	0.01	0.02	0.5	0
BREAD	0010	77	0.4	8.3	0.0	0.0	0	3	0.0	0	0	17	0.00	0.00	0.1	0
EGG	0064	110	7.1	8.2	1.5	0.0	51	120	1.0	164	93	691	0.05	0.17	0.0	0
FRANKFURT	0060	185	7.5	16.5	1.0	0.0	4	79	1.1	660	132	0	0.09	0.12	1.6	0
MILK	0448	264	18.8	8.9	26.8	0.0	640	501	0.4	273	784	358	0.17	0.94	0.4	4
ORANGE	0220	103	1.5	0.4	26.4	1.1	94	37	0.4	2	453	440	0.22	0.08	0.8	99
BANANA	0175	148	1.9	0.3	38.8	0.9	14	45	1.2	1	647	332	0.08	0.10	1.2	17
CHICKEN	0090	211	29.3	9.1	0.9	0.0	13	212	2.0	0	0	126	0.06	0.36	6.3	0
SHORTCAKE	0160	0	59.2	0.0	0.0	0.0	0	0	0.0	0	0	0	0.86	0.36	5.9	0
STRAWBERRI	0125	136	0.6	0.2	34.7	1.0	17	21	0.8	1	140	37	0.02	0.07	0.6	66
HAMBURGER	0090	257	21.7	18.2	0.0	0.0	9	174	2.8	42	405	36	0.08	0.18	4.8	0
BUN	0056	166	4.5	3.1	29.6	0.1	41	47	1.0	293	53	0	0.15	0.10	1.2	0
FR. FRIES	0128	350	5.5	16.8	46.0	1.2	19	142	1.6	7	1091	0	0.16	0.10	3.9	26
TOTAL		5528	263.1	348.7	419.1	12.2	2138	3043	29.5	5543	8627	17774	3.25	4.79	56.4	397
RDA		1800	44.0				800	800	10.0			4000	1.00	1.20	13.0	60
DIET STATUS		3728	219.1				1338	2243	19.5			13774	2.25	3.59	43.4	337
% OF RDA		307.1	597.9				267.2	380.3	295.0			444.3	325.0	399.1	433.8	661.6
% TOTAL CAL			17.9	53.4	28.5											

Figure 3

Journal
Administrative Offices
HP 3000 International Users Group, Inc.
289 South San Antonio Road
Los Altos, California 94022
U.S.A.

BULK RATE
U.S. POSTAGE
PAID
Permit No. 656
Los Altos, Calif. 94022