

Since the purchase of this system in July, 1970, advances in computer technology have substantially reduced the cost of both computer and peripheral devices to the point that the current duplication cost should be around \$25K. By the same token, a small dedicated system of restricted capability may be obtained for less than \$25K today. The greatest disparity, however, arises in consideration of the manpower resources required. A single competent engineer/operator can certainly satisfy the requirements of a dedicated system user if electronic interfacing requirements are limited. Although hardware effort may be assumed to be equivalent, the multiprogrammed time-shared system presents quite a different picture because of the sophisticated software involved. To date, the subject system has incurred approximately 8 man years of system programming. Three years ago, sophisticated systems software for small or medium size computers was the exception rather than the rule. Today most manufacturers have fully developed monitors and executives that have undergone considerable use and improvement. These systems must be adapted to the specific problem, however, and it is likely that a system

similar to that in NIMH could be developed today with less than 6 man years of system software effort. Development of such a system should not be attempted without this resource.

A series of small computers attached to a very large time-sharing system is another approach to laboratory automation. Its major drawback is that a very large computer system must be available. The approach discussed can stand alone without the support of a large facility. It has the added advantage that should a large computer facility become available, the shared computer can make use of it.

There are several alternatives available for the automation of a psychology laboratory. One approach is for a group of researchers to pool their resources and share a time-shared computer which is capable of servicing their real-time needs. This approach has been implemented at NIMH and is workable. A major benefit which is unique to the medium size time-shared computer approach is that much of the problem can be solved on a single computer. If more than one computer is required, this approach provides an easy to use interface to the other computers.

Behavior Research Methods & Instrumentation
1974, Vol. 6, No. 2, 146-148

Design and organization in a computerized psychology laboratory*

JERRY C. FORSHEE†

Indiana University, Bloomington, Indiana 47401

This paper presents six principles of laboratory organization and design. These six principles are an outgrowth of the past 7 years' experience in the Mathematical Models Laboratory of the Psychology Department at Indiana University. Each of the six principles is formally stated and then followed by explanation and examples.

INTRODUCTION

The several principles of organization and design for a computerized psychological research laboratory, presented here, are the result of several years of experience. The Mathematical Models Laboratory of the Psychology Department at Indiana University has been in operation since 1966. The laboratory computing facility is centered around an IBM 1800 Process Controller, the configuration has grown to include the CPU with 32K of core, 12 levels of priority interrupt and multiple process I/O points, several standard peripherals, a card read/punch, a line printer, a dual disk

drive, and several typewriters. Other peripherals on the system include two computer terminal facilities consisting of a CRT terminal, a character printer, and an incremental digital tape drive. This computing facility is used to support four time-shared subject running rooms. The research community using this facility may include, in a given semester, as many as three or four full time faculty and from five to ten graduate students.

THE PRINCIPLES

I. The Principle of Accessibility

The principle of accessibility states that information about the current state of the computing system must be immediately available to the user.

This principle demands a facility such that if the user needs to know the values of Core Locations 1-50 he needs only to enter on a terminal the command DUMP

*The research facility described herein and from which these ideas evolve is supported by Public Health Service Grant PHS R01 MH 16817.

†Requests for reprints should be sent to Jerry C. Forshee, Department of Psychology, Indiana University, Bloomington, Indiana 47401.

CORE 1,50, rather than write a program that calls a subroutine which will dump these same core locations.

To implement this principle, we have developed a series of special commands to be used interactively from a terminal. These commands facilitate system monitoring and enable us to modify the flow of activity in the system. Through the use of these special terminal commands, the user can perform system recovery procedures, obtain information about the current status of the system, or perform functions useful in debugging real time programs. All of these activities can be done interactively in real time without interfering with other operations in progress. By using clear mnemonics, by listing the vocabulary of programs at the terminal, and by tailor making the programs, we have greatly facilitated access to the system.

II. The Principle of System Attributes

This principle states that apparatus supporting real time activity should be an attribute of the entire computing system, not dedicated to just some portion of the computing system. In early years of the laboratory, we developed several laboratories, each with a fixed and narrowly defined configuration. For example, one laboratory consisted of a small theater with a random access slide projector; a second contained TV monitors. When apparatus is dedicated to only a portion of the computing system, a situation arises where all the apparatus needed for an experiment is currently unused but unavailable because it is dedicated to some aspect of the system that is in use.

We have employed this principle in the centralization of our stimulus presentation equipment. We have several devices that are used for presenting stimulus information to Ss that produce a video signal. These include four memory raster display devices. These devices take as input high speed parallel data from the computer and produce a video image for a standard television monitor. The information displayed on the monitor consists of the standard set of alphanumeric characters.

For these video devices one design scheme would be to allocate one memory raster display device to each experimental S running room. In such a system, it would not be long before some bright graduate student had the idea to run an experiment where he required two memory raster display devices in a single experimental S running room. To implement this, some amount of system rework would be required.

In applying the principle of system attributes, we have made these stimulus devices attributes of the entire laboratory complex, i.e., the total system. These devices are designated to the software by logical unit number, not by a numbering scheme mapping to the experimental rooms. Thus, the laboratory manager may, for our example, assign memory raster display devices with Logical Units 3 and 4. The hardware implementation of the assignment is simple. Each experimental room has several video channels and they

are all connected to a video distribution panel. The input to this panel is the set of video lines from the several devices generating such signals. Thus, the bright graduate student in our example need only turn the selector switch for his experimental room's particular video channel to the video source position for the display devices he has been assigned: Logical Unit 3 on the first channel and Logical Unit 4 on the second.

III. The Principle of Changeability

The principle of changeability deals with assigning, connecting, and documenting all process I/O points. Changeability demands that the assignment of process I/O points is well documented and that a change in assignment is easy to implement. Changeability requires that multiple sets of assignments are possible, and that it is possible to change quickly and easily between them.

To meet the requirements of the changeability principle, we have used a large MAC panel system through which all our process I/O lines run. This system consists of two large banks of terminal strips. Each bank of terminal strips has one side of each terminal strip connection point running to the MAC panel. The first bank of terminal strips has its other side used to terminate the hundreds of wires coming from the laboratory. The second bank has the other side used to terminate all the process I/O lines from the computer. Thus, with one jumper wire on the MAC panel, it is possible to connect any one wire from the laboratory to any one process I/O point. In the time it takes to insert a new MAC panel into position (a few seconds), an entirely different assignment of all the process I/O points can be implemented.

This flexibility allows the testing of new wire schemes for the laboratory in small stages, each stage being checked out before the next is begun. Also, during one experimental hour we may use a laboratory with one set of assignments. Then when the Ss are finished, we change MAC panels. In the next experimental hour, in the same laboratory room, we run an experiment completely different in its assignment of process I/O points.

IV. The Principle of Testability

This principle requires that it be possible, without the construction of a specific interface, to connect apparatus for the purpose of evaluation.

Our Utility (UT) Lab allows such testability. This facility allows immediate evaluation of a new idea to determine if it is feasible. Recently we were curious to know if pigeons could perform a task similar to a serial pattern learning task our human Ss had been doing quite successfully. We ran a cable from a pigeon changer to our UT Lab. This cable was terminated, jumper wires were placed on a MAC panel for the UT Lab interface, and the hardware work was completed. We ran a squad of pigeons for several weeks and decided that the project should not be continued. Thus, an entire new

application was implemented and evaluated all without need for new interface equipment.

The UT Lab interface consists of four units: (1) the user test panel (this panel provides a set of switches and lights and may be used to simulate hardware, or used in real time to input data to programs or to output status information, or it may be used to modify the wiring of the interface itself); (2) a set of standard computer interface modules (these modules provide the necessary electronics to allow the connection of many different types of devices, including: electromechanical, switch closure and solid state); (3) a bank of terminal strips for the termination of several sets of user cables; and (4) an interconnecting MAC panel (by using a MAC panel, the user makes the hardware wiring specific to his needs. All the lines from the user test panel, the conversion modules, and the user termination area along with power and ground lines run to this interconnecting panel. Thus to program the interface for a specific task, the user need only place jumper wires on the MAC panel).

V. The Principle of Administrative Dynamic Allocation of Resources

The principle of administrative dynamic allocation of resources requires that all unused resources be pooled, and that these resources are removed from the pool and assigned to a particular research problem only as they are needed and only in the amount they are needed. These same resources are returned to the pool immediately when their use is complete.

In applying this principle in the laboratory, we assign to an experiment the needed features of the hardware and software system, and only for as long as they are required. This system is in contrast to an "ownership" system in which resources are divided up, each being assigned to a particular experimental location.

This principle may prove very useful in a situation where the real time computing facility is a departmental facility. In such a case, the number of users is large but not all of them require resources at the same time.

We have applied this principle to our use of the software system's laboratory COMMON area, where a process program must store information about the experimental process. This COMMON is the only area of memory that is unchanged by the system across the many loadings and executions of the process programs during the running of an experiment.

We consider this laboratory COMMON as a pool. As each new experiment comes onto the system, it is assigned as many words of COMMON as it needs, and when an experiment is finished, the COMMON it was using is returned to the available pool and may be reassigned at any time.

VI. The Principle of Human Engineering

This principle demands that the computing facility be usable to the people for which it is established as a research tool.

Perhaps this is the most important concept presented here. No matter how well the hardware and software systems are designed and implemented, it will be wasted effort unless the organization also includes those features which allow the system to be fully used by the researchers. If the researcher cannot determine the whats, the whys, and the whens of executing his experiment, the laboratory does him no good.

Here are some specific examples of human engineering: (1) Not only should adequate documentation be available but it should be in the right place. The best location for information is where it is used. The best location for the computer cold start (power up) procedure is posted next to the computer console where the operation is performed, not in a manual in some office. (2) The best place for the scheduling of laboratory running times is not in a notebook in the laboratory manager's desk, but on a large blackboard located centrally in the laboratory where it can't be missed. (3) The best location for error recovery manuals is next to the device that lists error messages. (4) The best location for a summary of the job control language statements is next to the device that is used to produce them, i.e., keypunch, CRT terminal, keyboard. (5) A programming work space is needed where system documentation is kept absolutely up to date. Here programmers and programming teams can work in a studios atmosphere on those inevitable programming bugs.

When all of these principles are implemented in a research laboratory, the product is a scientific tool of great flexibility, which has the capacity to respond quickly to unique experimental requirements.