SESSION VIII

THE ON-LINE COMPUTER LABORATORY: FROM DATA COLLECTION TO DATA ANALYSIS: A SYMPOSIUM

ADAM REED, *New School for Social Research, President*

Interface systems in behavioral research

JUERG ELSNER and RUEDI WEHRLI

*Institute of Toxicology, Swiss Federal Institute of Technology
and University of Zurich, Zurich, Switzerland*

The methods and criteria to be used for the design of a general interface system are evaluated in the light of the number, variety, and variability of the data points in behavioral experimentation. A simple and flexible interface scheme is presented that should satisfy the needs of a behavioral research laboratory.

The quality and interpretative value of an experiment is often sensitive to the choice and number of simultaneously measured and controlled variables. This is the main reason for using a computer for behavioral experiments: having a central device that can monitor and protocol, on a single medium, a great number of experimental variables, as well as having rigorous control over as many parameters of the experimental environment as possible. The speed of a computer also allows one to monitor and control simultaneously a large number of experimental units, which can differ greatly in their structure and data characteristics. Thus, to fully use the potential of a computer, it is vital to have the capacity of a large number of connections between the experiments and the computer. In view of the variety and variability of the data used in behavioral experiments, a highly flexible data transmission system would be of great value. One requirement for such an interface frame should be the generality of its design, so that the exact nature of the data to be transmitted can freely be selected and modified with minimal effort and cost.

The aim of this paper is to elaborate criteria and methods for the design of a general data acquisition system and a control scheme suitable for the needs of a behavioral laboratory. Of the many different aspects of this problem, only one will be considered: the data transfer between the I/O bus of the computer and binary data points at the experimental site, irrespective

of the direction of this transmission (input and/or output). The nature of the data points shall not be specified: It may be binary information of any kind in itself or part of a binary coded analog value, instruction, address, or the like. The electronic characteristics of these binary data should be standardized within the system and may be converted into the proper format by special electronic circuitry. As an example, the design of a general interfacing scheme will be presented.

COMMERCIAL SYSTEMS

The present section outlines three main categories of commercially available interface configurations. The most straightforward, but also the most expensive, approach to an interface system is the use of the I/O bus of the computer (Figure 1). Besides costs, there are other

Figure 1. Example of an interfacing scheme using commercial interfaces and special electronics.

This work was supported in part by the Swiss National Science Foundation. We thank Richard A. Martin, Indiana University, for his helpful advice and cooperation. The second author is at ALWENA Electronic Design, Zollikon, Switzerland.

disadvantages to this approach. The I/O bus, not indefinitely expandable, should be reserved for interfacing specific computer devices such as terminals, magtapes, disks, line printers, and so on, for which purpose it is designed. This type of solution is also too centralized for most situations: The I/O bus is an integral part of the computer itself and as such may not be physically extended too far from the computer. If the experimental sites are not in the immediate neighborhood of the computer, long cables are needed to connect each data point to the I/O bus. Commercial interfaces group a number (e.g., 16) of bits of one specific kind together into one unit, and this generally does not conform to the actual grouping among the experimental stations. A highly complex data structure would be the consequence, the bits of one I/O word being distributed among a number of experimental units, and one station being composed of bits of a number of different I/O words. This leads to obscure and tortuous programming. Special electronic circuits have to be built in most cases for converting available data characteristics to the needed data format. Such a solution is only appropriate in highly specialized applications with very few data connections needed, a situation that is beyond the purpose of this paper.

Other commercially available solutions are constituted by industrial interface systems such as CAMAC (Figure 2). These make better use of the I/O bus and allow for more favorable (but still not ideal) cabling solutions, by using remote versions of these systems. But some of the objections made to computer I/O-bus interfacing are also applicable here: The grouping of the data is made in a similar fashion; generally, greater variety of sensing and output data formats exist, but they also have to be completed with specially designed electronics for driving high-power lamps, solenoids, sound generators, etc.

Interface systems specially designed for behavioral laboratories are usually limited in expandability and flexibility (they allow a number of stations, in the order of about 10 maximally, with about the same number of

Figure 2. Example of an interfacing scheme using local (left) and remote (right) I/O controllers.

Figure 3. The "bus"-type data path, where the data points, located in separate stations, are connected together by a "daisy-chained" communication bus.

Figure 4. The "multiplexer"-type data path, where the data are collected and distributed between the data points and the I/O bus of the computer through a controller.

controlled stimuli each) and are very expensive. They are confined to special computer types, so that the criteria relevant to the selection of a computer may strongly interfere with those concerning the interface system.

The most general, and perhaps the best, commercial approach to our problem may be an industrial interface system; however, a strong argument against such a system is that for most applications special electronics must be built anyway.

BASIC COMMUNICATION TECHNIQUES

This section will classify and discuss basic communication techniques according to the following characteristics: (1) data path structure, (2) transmission type, (3) transmission control, (4) data identification logic, and (5) process control. These considerations will serve as guidelines for the development of an interface scheme for a behavioral research laboratory described in the next section.

Data Path Structure

Two basic techniques can be adopted for the data path structure: (1) the "bus" type (Figure 3), where each data point is logically connected to one unique "daisy-chained" data carrier, and (2) the "multiplex" type (Figure 4), where the data are distributed and

collected by a special controller through as many data carriers as there are data points. The two techniques can be combined to find an optimal design of the data path structure in matching it directly to the logical structure of the laboratory to be equipped. A conceivable design could be the distribution of the data by a controller to a number of different laboratory units where individual experimental stations are connected together by a common bus. Each station would have its own small controller, distributing the data to the electronic conversion logic. The data collection would work along identical paths. An ideal interface system should be flexible enough to allow any kind of adapted communication structure to be built.

Transmission Type

Two basic techniques of information exchange can be incorporated: (1) parallel transfer (Figure 5), in which a number of data carriers transmit simultaneously one binary information (bit) each, forming together one word of information, and (2) serial transfer (Figure 6), in which a binary word is transmitted along one carrier in a bit series according to a predefined time sequence. These two basic schemes can be combined in a synchronous transmission of bit series along a number of parallel carriers. The expected data rates and the number of data points to be transmitted for a given data transfer determine the transmission type to be chosen. Serial transmission takes considerably longer than the CPU cycle time; thus, programming is rendered more cumbersome by the fact that between two data transfers the software driver has to wait for completion of the first transfer. Events in behavioral experiments may be slow compared to CPU speed and to the transfer speed of a modern serial interface (about 50,000 bits/sec or more). However, if many stations are connected to the computer (e.g., over 30, a number that can easily be controlled by a modern minicomputer with fast software), momentary data rates may get quite high. The high capacity of the CPU should thus not be reduced by a slow transmission speed. Still, serial transmission may have decisive advantages, if wiring costs get determinant for connection distances of more than about 50 m or if standardization with other kinds of data transfers (e.g., computer to computer or computer to terminal, etc.) is the goal. However, for any other application with a

Figure 5. The parallel data transfer.

Figure 6. The serial data transfer.

large number of data points, parallel data transfer at CPU speed with the number of parallel data carriers matching the computer word size in bits seems to be more economical and more easily handled by the software.

Transmission Control

Two basic schemes may control the data transfer: (1) the "handshaking" method, in which the receiver acknowledges each data transfer to the sender, which in turn releases the data or sends new data only upon receipt of this acknowledgement; (2) the "master-slave" method, in which the receiver is considered to be ready for data reception at any time and needs a predefined lapse for data acceptance, after which the sender may release the information. Handshaking needs special control logic and cables, while a master-slave communication is simpler to design. However, it is not always possible to have a master-slave relationship between sender and receiver. The most striking example is the computer considered as receiver. Since the computer may be busy with another task when it should receive some data, it may not be used as slave, except perhaps if direct memory access (DMA) is selected. As the CPU is not involved in a DMA data transfer, this technique should not be used for behavioral experiments in which the computer has not only monitoring but also control functions to perform. Thus, an interrupt and acknowledgement technique must be incorporated for the data transfer from the experiments to the computer. On the other hand, for simplification of the design, the other direction of data transfer (the output) may be constructed as a strict master-slave relation without major loss of flexibility.

Data Identification

Two basic types of data identification structures may be considered: (1) "data type identification," in which

data of the same kind are grouped together and distributed to different stations after being converted, and (2) "station identification," in which all data of a given experiment are grouped together. The first solution is the one adopted by the first two commercial interfaces described above. There is an apparent economic advantage of grouping, for example, 16 channels of identical electric characteristics together on one big board, but it may be more than outweighed by the poor data structure and the cabling efforts to be performed. For most cases it is better to have small units on which no more than four or, for special applications, even fewer signals are processed. A high degree of decentralization can be achieved by building for each experimental unit one small controller which receives and sends the data as standardized binary code and performs the conversion between the physical signals and the binary code.

Process Control

Two basic concepts can be adopted for process control: (1) central control, where the computer makes all decisions about scheduling of the data, and (2) local control, where microprocessors relieve the computer of the process control. The computer is then involved only in the storing of the experiment history and more basic decisions, like the start and end of an experiment and the choice and definition of the schedule. Data throughput and decision-making complexity, as well as the number of serviced stations, and the number of additional tasks asked from the computer determine if the use of microprocessors is adequate. An interface system should not exclude the possibility of incorporating distributed computing by microprocessors.

A BEHAVIORAL INTERFACE SYSTEM

An interface system was developed for a laboratory of behavioral toxicology. It shall be described here as an example of a flexible communication scheme. A prototype system is operational and has successfully been used in a series of experiments of a maze with 18 optical gates and an operant chamber with 15 different stimuli and two levers. The control logic has been designed so that it is only a small part of the total electronic circuitry and can easily be adapted to most kinds of applications.

The computer, selected by a separate evaluation not to be described here, is a PDP-11/34. The laboratory architecture is relatively compact, necessitating connections no longer than about 30 m. The laboratory equipment is expected to grow over the next years, leading to a considerable increase in the total data transfer rate. The system philosophy includes most applications. New experimental set-ups should require mainly replications of existing electronics.

Interface Characteristics

The basic characteristics of this interface system can be described as follows (Figure 7): Each experimental

Figure 7. A general interface scheme for a behavioral research laboratory.

station is treated as a logical unit with its own controller housed in a 19-in. rack dimensioned for 100 x 160 mm Europe cards. It contains a power supply and a simple standardized I/O logic for communication with the computer through 16 parallel channels (the word length of the PDP-11 is 16 bits). A unit can be equipped with input or output logic, or with both. The data conversion logic is incorporated into the same rack. Residing on Europe cards, this logic can be connected to any input or output bit (or a combination of bits if needed), supported by the I/O logic at an internal data bus. This connection is made through opto-couplers for security purposes. Therefore, any kind of input and output logic can be built with modules of integrated circuits, which are available today in great variety, and connected to the opto-coupler input or output.

On principle, the data path is structured as a data bus connecting the PDP-11 UNIBUS through a parallel 16-bit general device interface (DR11C) to one station after the other in a daisy chain. However, it is easy to design a switching station which multiplexes the data channels to different groups of daisy-chained units. Also, the unit controller can be modified easily to service two or more single experimental stations.

The communication scheme works as follows (Figure 8): For the output, the first 16 bits (the first word) sent by the computer through the DR11C contain a 7-bit station address, plus 1 bit of address flag, which has to be set to one to indicate that the word contains

Figure 8. The communication scheme for the interface system.

an address. This word can contain meaningful data in the other 8 bits. The address is decoded by all controller units on the line, and the one to which the address corresponds accepts these data. The time required for decoding of the address and the data reception is shorter than the time needed by the computer to send the next output word. Thus, the unit receiver is always ready to accept data and is designed as a slave. If 8 bits of data contained in this first word are insufficient, the next words (having their address flag clear and sent in a predefined sequence and number) contain the additional bits of data, all directed to the same station that was addressed by the first word. Only the transfer of a word having the address flag set redefines a new receiver. For every data transfer, constituted by a certain sequence of words sent to a unit, the complete status information

will thus be transmitted, irrespective of whether a specific status is changed by the operation. However, any kind of logic can be defined by codifying the instruction to be taken by the unit controller. Since an essentially unlimited number of data words can be transmitted by this scheme to a unit, it is possible to download a microprocessor memory, which may be incorporated into a controller.

The input functions similarly but, in addition, an interrupt logic controls the use of the input bus. At a high rate (2 MHz), each unit in a group of maximally 16 units is enabled, in turn, for input by a logic contained in one of the controllers. If a greater number of units must be connected or more groups are needed, the same procedure can distribute the control of the bus among the groups at a higher level. If an input is requested, the cycling of the units is stopped, and an interrupt sequence is initiated at the computer, followed by the data transfer of the predefined number of words.

The system was built in CMOS at 12 V. The maximal length the bus may attain has not yet been precisely evaluated, but the length of 15 m produces no distortion of the signals.