CONCLUSION

You have now heard and seen the specifications of the SCAT language. We should go on and discuss the interface hardware of the SCAT system because many interesting ideas are contained in it. But I thought that, given the limited time available, you would like to hear first about the language. If you are interested, we can talk hardware during the discussion period.

In conclusion, I hope you will agree with me that the SCAT system is a significant jump forward on the path of experimenter computer interaction. The SCAT system has been designed to simplify enormously the problems enumerated at the beginning of this paper, or to eliminate them altogether. I look forward to your questions and comments.

REFERENCES

- CIOFALO, V. B., TEDFORD, R. H., GOLDBERG, M. E., & DODGE, D. L. The use of a digital computer for control of behavioral research in animals. *Behavioral Science*, 1968, 13, 252-256.
- HABER, R. N. An on-line computer in a visual perception laboratory. *Behavior Research Methods & Instrumentation*, 1968, 1, 86-93.
- MARKOWITZ, J., & NICKERSON, R. S. On timing events in computer-controlled experiments. *Behavior Research Methods & Instrumentation*, 1968, 1, 82-86.
- UTTAL, W. R. "Basic Black" in computer interfaces in psychological research. *Behavior Research Methods & Instrumentation*, 1968, 1, 35-40.
- UTTAL, W. R. *Real time computers—Technique and application in the psychological sciences*. New York: Harper & Row, 1968.

Can a small dedicated machine find happiness in a company that pioneered time-sharing?¹

JOSEPH MARKOWITZ, BOLT BERANEK AND NEWMAN, INC., Cambridge, Massachusetts 02138²

In the context of an automated psychophysical laboratory, the adequacies and inadequacies of a small computer are discussed. Attention is given to the rationale for choosing the implementation in particular a dedicated small system as opposed to time-shared use of a larger system. The variables discussed include cost, flexibility, language level, and storage capabilities in addition to reliability. A compromise position that appears most viable for the future is also suggested.

Several years ago, we had the opportunity to build a new laboratory facility for psychological research. Specifically, the laboratory was to be used for basic psychophysical research with acoustic signals. In certain instances, the implementation choices may have been closely tied to the portended nature of our work. In general, we think they were not. Indeed, with small modification, the same facility now serves a much broader range of psychological research interests. Perhaps such versatility owes its due to the way in which the facility was, in fact, implemented. For various reasons, which are the very point of this paper, we decided to automate our laboratory. The medium we chose was a small, dedicated, general-purpose digital computer—the PDP-8.

Our need for automation arose, we felt, because of the drudgery that defines psychophysics. Our psychophysical experimentation is characterized by a simple, repetitive, short sequence of events where all the alternatives are drawn from a small closed set. It is characterized by a need for numerous observations in each of a few fixed conditions. The operations are routine and boring. Errors made by the E as to what he presents (or did present) are indistinguishable from errors on the part of the O as to what he observed. In this sense, errors are very costly.

Psychophysics, then, does not demand a computer of high intellectual ability or blinding speed. Its tasks are not that challenging. Rather, psychophysical research requires a tireless and accurate drone.

Quite simply, precision increases as E errors decrease, and as the number of observations increases. Unfortunately, the rate of an E's errors accelerates as the number of observations increases. On the other hand, a computer can keep its own error rate satisfactorily and constantly low.

In designing our laboratory facility, we were confronted with several options. Because all of these options are nominally open, even today it is worthwhile reviewing them.

The first option we examined was construction of a special-purpose facility out of logical building blocks. This was, of course, a proven approach. We, ourselves, had had experience with three generations of logical building blocks. We were intimately familiar with the clatter and electromagnetic splatter of relays, the shocking and geriatric nature of vacuum tube logic, and the unforgiving frailties of transistors. We knew, too, the joys and sorrows of a variety of connectors and programmable patch panels. Finally, we knew the disproportionate number of logic modules and readout devices required for even the simplest of data analyses. All of these lessons stood us in good stead, we felt, and we never regretted the experiences of the past. Yet, we were firm in our resolve to depart from the past, for the lure of computers was irresistible. The printout was on the wall.

Another way in which we could have accomplished the automation of our psychophysical laboratory was to use time-sharing facilities available to us. We could conveniently have run our experiments on one of several operating systems. Our colleagues interested in running manual-control experiments were setting out to do precisely that. Moreover, ruling out such an alternative would be particularly significant in view of our past history with time-sharing and with computerized psychological experimentation.

Bolt Beranek and Newman Inc. (BBN) has actively pursued time-sharing system research, development, and application for nearly a decade. Historically, we developed one of the first operational interactive time-sharing systems in 1962. That system was implemented on a DEC PDP-1 with the support of the National Institutes of Health.

A succession of hardware and software improvements enabled us to move from 4 to 64 simultaneous users and to increase the number and diversity of languages available to the individual user. In 1963, development was undertaken on a second PDP-1 interactive, time-sharing system. This system provided a number of operational programs in a specialized language at remote terminals in the Massachusetts General Hospital. In 1966, we implemented a real-time hybrid I/O interface to our time-shared SDS-940. This interface is called a Hybrid Processor, and is capable of accurately timing real-time data transfers between

real-world devices and the core memory of the SDS-940 without CPU intervention. We have used our Hybrid Processor to control an on-line video input device, to sample and generate speech waveforms at up to 33 kHz, to interface to a CRT display, and to control a hybrid analog computer (Applied Dynamics 4), to name a few.

We have also designed a scheduler that will handle a number of synchronous and asynchronous real-time processes. The scheduler will permit a process to get on the system only if the system can guarantee service to the real-time demands of this and all other processes. Copies of both the scheduler and hybrid processor are presently being installed in our most recently acquired time-shared system, a dual-processor PDP-10 computer. The system is backed up by a 100-million-word disk for bulk storage, and has been modified by hardware "paging" of our own design.

As you can see, quite a bit of hardware and software development went into each of our time-sharing systems to provide adequate real-time service. This, of course, is precisely the need for controlling psychological experimentation. Nonetheless, we decided against time-shared service for our laboratory. This is ironic, indeed, because among the very earliest of computer-controlled psychological experiments were some done by BBN on the very computer on which we pioneered time sharing (Swets, Millman, Fletcher, & Green, 1962).

The option we did exercise was to base our psychophysics laboratory around a small, dedicated, general-purpose computer. A computer-based system turned out to be cost-competitive with the build-it-yourself option at the time we made our initial decision. If anything, the economic balance has swung even more markedly in favor of the computer. Of course, this had not been the case when manufacturers offered for sale only medium to large computers. Now, however, small machines are a proven product line. As a consequence, we think that the specially designed collage of logical building blocks must be forever foreshadowed.

Building up special-purpose logic means, almost by definition, physically disparate pieces for each conceptual function the equipment must perform. With even a few functions to be implemented in this fixed way, the amount of logic required exceeds that necessary to build the processor of a small general-purpose computer. Moreover, core storage costs, even at present, are about \$2.00 per word, a figure to be compared with the cost of a three- or four-digit electromagnetic counter. Perhaps it is best to think of the general-purpose digital computer as simply a particular, concise arrangement of some hardware.

By opting for a general-purpose digital computer upon which to base our facility, we also gained a degree of flexibility unobtainable in a specially built piece of gear. But, of course, as much flexibility and more would accrue if we chose to be a time-shared user of a large computer system. And yet we vetoed this course of action. We vetoed it on almost religious grounds, but we shall try to detail our reasons in as rational a fashion as possible.

Basically, our choice of a stand-alone system rested on the fear of unreliability of a large, multi-user system. We were then and are still of the opinion that unreliability is inherent in so complex a system. There are simply many, many more pieces, hardware and software, each with its own chance to fail.

A second consideration related to, but still distinct from, the reliability of a large multi-user system, is the integrity of one's own portion of the system. How secure is any user from the malfeasances of other users, whether of malicious or benign intent?

A third consideration was our feeling that the real-time demands that we would place upon a time-shared system in certain kinds of experiments were basically incompatible with the system's desire to schedule its activities "efficiently."

Let us discuss these three points in turn.

Reliability. It is important to understand for ourselves, and to communicate to others, the premium we place upon reliability. Architects of large computer systems and directors of large computer centers rail at our demands for reliability. Generally so tolerant of our meager demands on their vast reservoir of compute-power and storage, they cannot understand our phobia of system malfunction. But our concern is hardly groundless, our fear scarcely irrational. The problem is that our Ss-people, animals, or whatever—change over time. They learn; they forget. They get rusty, or they rest up. The problem also is that our experiments, by design, place emphasis upon changes over time. All of us, I'm sure, can think of experiments where equipment malfunction on a given day may jeopardize an entire experiment. The cost, then, may not be simply the actual time during which the system was inoperative, but all of the experimental hours that preceded the failure as well. And, too, when experimental hours are lost, they may involve a number of people: several Ss, an E, and so forth. We also run the risk, if malfunctions are frequent, of tarnishing the image of omnipotence of the E in the eyes of his Ss. And finally, even though the E may be able to excuse the equipment for malfunctioning, the Ss may not be as gracious.

Most recently, informal sources indicate that an upswing in the reliability of large time-shared systems may be in the offing. The idea seems to be that a portion of the potential savings in collectivity will be used instead to provide for a degree of parallel organization or redundancy in the system. The premise is that such organization will allow the system to provide fully adequate service to a decreased number of users in the event of some system malfunction. Certainly, the concept of parallel redundancy to ensure some minimal level of functioning is a proven one. It is not at all clear, however, that a degradation over all can be traded off against a reduction in number of users. Besides, the limits to which this approach will be pushed will be determined by what standards of reliability the majority of users demand. As we have argued, however, the psychologist anxious to have the computer control his experiments may be rather more demanding than most.

Suppose, however, that degradation can be made to manifest itself solely as a decline in the number of allowable users. Who is it that gets "bumped"? When the system begins to falter and can accommodate fewer users, how much bargaining power does the psychologist have? The psychologist can keep the omnipotent CPU of a large system busy for the merest fraction of time. What law of economics dictates that he may pre-empt the inversion of a grotesquely large matrix? Paying a price out of proportion to the service must ultimately be the result.

Somewhere lower down in the hierarchy of reasons underlying our rejection of the time-sharing approach was the relative dollar cost of the two alternatives. According to standard accounting practice at then current pricing, the two alternatives were remarkably similar in cost. They still are. They may always be. Not that the equation of costs is axiomatic necessarily. More likely, those who engineer time-sharing systems reinvest whatever savings there are to be had in pooling of resources and needs. The savings are plowed back into greater capacity, compute-power and storage, as well as additional peripherals. The savings may also serve to underwrite more powerful languages, and a more intelligent time-sharing "executive," all of whose costs are ultimately borne by the overhead of the system anyway. Naturally, constraints operate to influence the decision of what savings should be rebated to the users directly but these are largely covert, tempered only by what the traffic will bear. At present (and perhaps always), the traffic will bear up to, but surely not far in excess of, what a small dedicated machine would cost to own and operate.

Integrity. The security of a user in a time-shared computer

system is a concept of importance, not only to denizens of the nether worlds of big business or national defense. It is a concept of importance not only to those of us who exaggerate the importance of our work and the larcenous intent of our colleagues. For, if it is dangerous for our enemies to have access, how much more dangerous is such access in the hands of our friends? Presumably, people to whom we are neutral present an intermediate danger. Another maxim we might as well accept is that no realizable system can be totally secure. What compounds the problem in a time-shared computer system is the stratification of the users. All users of a system are not equal. The status of some is far above that of others. While most users have access to only their own pieces of the system, programs and data, others have broader access. Whatever safeguards are established, there are always a few who have keys that unlock all the doors. It is not clear that this need always be so, but at present it seems to be the case.

The differential status of various users works to the disadvantage of the small, real-time customer (for such is the psychologist, in the main) in other ways. System maintenance, and worse, system revision, are largely scheduled for the larger good. How strongly can one press his case to have always available the hours of noon till one o'clock particularly if that hour per day represents the sum total of use by the individual? Even worse, careful examination of the records might indicate that the hour of use reflects but several seconds of CPU time—according to some, a measure of the inefficiency of the machine use, and, by association, the unworthiness of the job.

Real-time scheduling problem. One of the considerations that distinguishes the experimental psychologist's needs from those of many other real-time users is the asynchronous nature of many of our demands. In many applications—signal processing, or simulation, for example—it is possible to predict in advance the exact times at which service will be needed. However, many of our experimental needs cannot be so easily predicted. "From a scheduling point of view," it is claimed (Fiala, 1968) "the hardest processes to handle are those which vary rapidly at some times, and slowly at others. The system should attempt to guarantee satisfactory service to such a process under peak loads, while simultaneously using left-over time efficiently when the process imposes a light load on the system."

In other words, the higher the interresponse-time variance, the more difficult the scheduling problem. But for certain problems of interest to the psychologist, events of note are approximately Poisson-distributed. And, it is well known that the variance of a Poisson-distributed random variable is equal to and therefore grows precisely as its mean. Is this to say that we psychologists who deal with the relatively slow response rates of flesh and blood present quite a difficult scheduling problem? The implication that the less demand we make on the system, in terms of a low response rate, the more difficult a scheduling problem we pose, is indeed an interesting one.

The heart of the real-time processing problem is the input-output portion, rather than the computation demands. The program of a user operating in a time-shared environment has three alternative ways to control input-output functions: directly, or according to a priority interrupt scheme, or by consigning them to a separate hardware configuration, sometimes referred to as a hybrid processor.

Direct control of input-output functions means reasonably synchronous interlacing of the input-output processing with computation. As a result, the blocks of time left over to share may be so small, relative to the program-switching speed, that time-sharing may be infeasible, or uneconomical, for moderately high-bandwidth input/output requirements given direct control.

The use of a priority-interrupt scheme allows independent

number scheduling of input-output and computation and, therefore, more efficient scheduling of computation may be implemented because, in general, computation is real-time insensitive. But, if enough devices are attached to the interrupt lines—and psychologists, likely, want to attach such things—then, status tests must be made—an inefficient situation. Moreover, the pigeon keys and people pushbuttons of the psychologists can hardly compete in priority with the system's tape, disc, drums, display, etc.

Most interesting of the alternatives is the one called the hybrid processor, which says that what is needed for efficiency is a separate hardware configuration for input/output processing, which is like saying that it needs a separate computer, which is like saying that one might just consider having a separate computer. The hybrid processor is provided with its own portal to the memory of the system, and, as we shall see, a reasonable proposal for the psychologist's own small machine is a similar portal to the large machine—where time is no longer so critical, inasmuch as the small machine can provide considerable buffering.

It is probably well to mention the rationale according to which participation in a time-sharing system is usually proposed. Basically, the argument is an economic one. Users might have need for a piece of hardware or software so infrequently as to be unable themselves to justify its cost. Yet the equipment is still desired. By pooling wishes and wants with others, these things may be acquirable. There is undeniable merit in the approach, but a danger as well. There is too often too little safeguard against acquiring pieces of the system that no one will use. Picture yourself coveting a novel piece of gear. Unable to justify it yourself, you argue that, while you have no real use for it, someone in the community of users must certainly. It is a little like buying your wife a sabre saw or your two-year-old a set of electric trains.

It would be nice for systems to keep statistics on the use of their components, hardware and software. This sort of "overbuy" behavior may be more common than not. In a sense, it may be somewhat unavoidable inasmuch as a system must be planned before all the users and uses can be exactly determined. The counterclaim, of course, is that the sort of thinking to which we are here objecting is precisely the guarantee of flexibility that will promote the use of the system. We only suggest that the cost may be high.

The system configuration at which we have arrived is really quite straightforward—a basic PDP-8 with a high-speed paper tape reader and punch at the heart of the system. Equipped with only 4K of core and no other bulk-storage devices, the system relies on paper tape as an auxiliary storage medium, and, as we shall see, occasionally needs several passes to collect data and perform rudimentary analyses upon it. Two digital-to-analog channels drive either a display scope, added only recently, or the transducers for our psychophysical experiments, e.g., earphones or vibrators.

Typically, three psychophysical Os are run in conjunction, and each has a four-button response keyboard convenient in that all Ss' responses are easily encoded as a single PDP-8 word. The Ss' response keys are also or-ed into the interrupt line along with the overflow bit of our 12-bit clock, driven by a 1-kHz ticker. This is illustrative of our solution to the PDP-8's lack of a priority interrupt structure—our interrupt handling routine is the simple one of reading in all external buffers.

Instructions, timing-control bits, as it were—and feedback information are displayed to the Os by means of indicator lights and form a single 12-bit buffered output word. A second 12 bits of indicator lights are provided for the E. Another 12 bits of control information serve multiplex switches that enable us to

gate analog signals to our Os. Finally, another buffered output word provides control bits for a 3-channel projection tachistoscopic projector.

Initially, we had no high-speed I/O device, but when we called heavily upon the reader and punch associated with our Teletype, we found a distressingly high incidence of failure. In our opinion, some sort of high-speed input-output is a virtual necessity . . . and this is equally true whether or not the system has bulk storage in the form of additional core, a disc, or drum.

There are several options as to high-speed I/O devices. Some are expensive—quite expensive in the case of multichannel digital magnetic tape, and somewhat less in the case of microtape. We adopted paper tape some 3 years ago when there was hardly any cost-competitive alternative.

Nowadays, paper tape is still appealing from the standpoint of price, as well as compatibility with other systems if that need should arise. However, two relatively recent alternatives are well worth considering; the first is an inexpensive mag-tape machine, the other is a high-speed communications link to some other facility.

The inexpensive mag-tape machine takes a bit-serial data stream and modulates it, thus making it compatible for storage on a good quality home tape recorder. Demodulation of the stored analog signal can subsequently reproduce the bit stream for reentry into the computer.

The high-speed communications link is perhaps the most exciting new avenue. Properly done, with the cooperation of the computer utilities available, such a link can effect the compromise that must surely come for the user who desires a dedicated small system yet yearns occasionally for the benefits of a big system.

The dedicated system could be free-standing during crucial experimental applications, yet store programs, data, and institute analyses on a larger, more suitable machine. Control may be considered to be with the small machine. A separate, conventional Teletype link to the time-shared system might then be reinterpreted as providing off-line file handling and programming capabilities.

REFERENCES

- HALLA, E. R. Scheduling of real-time processes in a time-shared environment. Unpublished Master's thesis, Massachusetts Institute of Technology, 1968.
- SWETS, J., MILLMAN, S. H., FLETCHER, W. E., & GREEN, D. M. Learning to identify nonverbal sounds: An application of a computer as a teaching machine. *Journal of the Acoustical Society of America*, 1962, 34, 928-935.

NOTE

1. This work was supported by the Human Performance Branch of the NASA-Ames Research Center under Contracts NAS 2-2676 and NAS 2-5108.
2. Present address: Computer Systems for Medicine, Boston, Massachusetts 02115