DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS 02139

To: W. B. Davenport

From: Ad Hoc Committee on Educational Computing Resources

Date: July 7, 1977

Subject: Proposal for a Department Educational Computer Facility

Two earlier position papers from this committee described a policy position that a department computer facility should be acquired, and listed requirements for such a facility. The committee has reviewed several possible implementations of a facility, and has narrowed the possible courses of action. This paper describes some of the considerations involved, and then recommends a way of proceeding.

The recommended action involves basically choosing one of two alternative implementations of an educational computer facility: a Digital Equipment Corporation PDP-10 using the TENEX operating system, and a Honeywell Level 68 using the Multics operating system. Technically, either system could satisfy our needs; economics strongly favors the PDP-10/TENEX choice. The Multics alternative is included because it is technically suitable, and the possibility of a gift should not be overlooked. (We are talking here of acquisition of a separate, department-controlled facility, not of use of the I.P.C. Multics service). Since acquisition of either of these systems requires first the acquisition of a large sum of money, we propose two alternative holding actions (corresponding to the two implementation possibilities) that allow immediate needs to be met.

As background to the proposal, we first review two general approaches to implementing a department facility that do not appear technically feasible--the mismatch with requirements is too great--and one approach that doesn't work very well. We then describe the approach that appears to be technically feasible, and the short- and long-term proposal that follows that approach.

An approach that doesn't work: the fleet of mini-computers

The first of the two unworkable approaches is with minicomputers. Two possible implementations were explored in some detail, the Digital Equipment PDP-11 and the Hewlett Packard HP-3000 series II, and it was apparent from those two that there was little point in examining others. The PDP-11 was the starting point since it appears to have essentially all the software required: a suitable operating system (UNIX), FORTRAN, APL, PASCAL, and LISP. Three difficulties were immediately apparent:

1) Each of the languages was developed by a different source for a different operating system; integration and maintenance would become our responsibility.

2) The user's address space of the PDP-11 architecture is too small. (64K bytes of data and 64K bytes of program, in the bigger versions of the PDP-11.)

3) Several--immediately 3 or 4 and ultimately perhaps 10 or more--machines would be needed, and software and hardware intercommunication at this scale are not available.

Although the first problem could potentially be coped with, the second is much more difficult to handle. The vision of having an impact on engineering education and preparing students for the world of the future requires that higher level languages with substantial supporting libraries and previously prepared ("canned") subject-specific software be the dominant mode of use of a department facility. For some users, the PDP-11-sized address space would work very nicely. But the programs and data involved in such a vision will, in many other cases, simply not fit into a small workspace. For example, the teaching staff that employ the present DELPHI PDP-11 system in 6.031 find that they must carefully devise homework that can be solved in a small workspace. Application to anything but toy problems does not work. (This effort is an example of an increasingly frequent occurrence with mini-computers: the expenditure of significant intellectual effort to fit into a limited addressing space; while memory is relatively cheap and available but unusable because of limits in the hardware architecture.) Extension someday to use interesting but large programs such as the MACSYMA algebraic manipulation system, even just for demonstration, would be impossible.

It is widely suggested that DEC will soon extend the PDP-11 architecture to allow a larger address space, but even if that extension occurred today, it is likely that it would take several years of software development before the new architecture is fully usable. Without software, the extension does not help meet the department's requirement.

Finally, the lack of network facilities would mean considerable local engineering or else running the ten or more systems independently, each holding one tenth of the community's files. The resulting partitioning would be an administrative burden and would inevitably lead to "breakage", in which some machines are overloaded while others stand idle.

A similar situation applies to the Hewlett-Packard HP-3000. 1) APL requires special microcode support, and to be effective should be the only user of a particular configuration. Also LISP and PASCAL are unavailable, 2) the user's address space is somewhat larger (many 16K byte segments) but much less flexible than in the PDP-11 (only 32K bytes of data space), and 3) no multisystem interconnection software and hardware is available. From limited knowledge of other mini-computers, these problems are universal, with the additional burden that one or more of our intended languages of use are not implemented. Thus we conclude that despite the applicability of mini-computers to an increasing range of situations, a department-wide educational computing facility is not one of them. The address space limits are just too severe.

Another approach that doesn't work: traditional operating systems

A second approach was examined and discarded: to use one of the standard time-sharing systems offered by traditional mainframe manufacturers: IBM's MVS/TSO or VM/370, Univac's EXEC-8, Burrough's MCP, Honeywell's GCOS, and CDC's SCOPE. The mode of operation envisioned for a department computer facility is essentially operatorless, more like a telephone exchange than like a traditional computer center. This mode is not only feasible for a system used exclusively in an interactive mode, it is essential to control expenses. But all of the traditional operating systems were developed first as batch operating systems, with tape drives, card readers, and printers, all of which require a staff of operators near the computer. As a consequence, these systems were designed with the assumption that an operator is always available. This assumption is usually embedded deeply in the system design, showing up in the form of elaborate, multistage system startups, inability to operate without a card reader and printer and magnetic tape drives, and dependence on an operator to respond to all kinds of system problems. When system crashes occur, these systems cannot be restarted casually; human intervention backed up with substantial experience is required to judge which of several subtly different restart methods is required. It is difficult to run these systems without a highly-trained around-the-clock staff of operators; the M.I.T. Information

Processing Center runs exactly such an operation. On the other hand, the department's DELPHI system as well as several of the dedicated research computer facilities provide some evidence that a properly designed system can be operated much more casually, and without the staff of professionals.

A second problem that tends to pervade the traditional system designs is that interactive use was an add-on rather than part of the initial design. As a result, operating algorithms and other system implementation decisions usually are based on an assumption that the interactive load constitutes a small fraction of use. When one attempts to run one of these systems under a predominently interactive load, they are stretched in ways the design is not prepared for: the most common result is wasted performance.

We therefore conclude that using any of these systems would be impractical, requiring a high budget for operations and probably also a substantial development cost to retailor the design to be effective in a completely interactive environment.

An approach that doesn't work very well: the status quo

A third approach, the one that will certainly be followed if no action is taken, is to continue with the present method of providing computing resources for department educational programs. This approach involves operating the DELPHI PDP-11 facility for subject 6.031, purchasing time for other educational uses at the M.I.T. Information Processing Center, and (illegally) "borrowing" time from various research computers. The department currently uses Multics for time-sharing service (6.030 and computer science laboratories) and the IBM 370/168 for APL and FORTRAN service for several E.E.C.S. subjects. This approach has several surprisingly high costs:

1) The 1976-77 department expenditure for educational use of computers was about $145K for the year, and is rising: (figures include both IPC billings and terminal rent)

6.030, 6.171, 6.176, 6.912 use of Multics	$ 90K
6.031 use of DELPHI	25K
6.301, 6.302, 6.073, 6.333, 6.202, 6.242 use of APL	25K
6.725, 6.201, 6.241, 6.432, 6.341, 6.291 use of FORTRAN batch service	5K
	$145K

2) The DELPHI system is currently stretched to the limit of its capacity by growing enrollments in 6.031, and is unable to absorb the load that would be imposed by proposed revisions in 6.031.

3) Enrollment is severely restricted in the software engineering, artificial intelligence, and performance measurement laboratories. The demand probably exceeds twice the 1976-77 supply of 70 places/year. For 1977-78, there is a hope that the supply can be increased to 150 places. That increase would require an additional expenditure of about $25K.

4) Innovation, especially in use of computing in traditional electrical engineering subjects, requires arguing for an increase in the budget; this requirement undoubtedly stifles innovation.

5) There is no way that a department student can legitimately use computational facilities for homework outside of those subjects that require it.* ("Anything that is not mandatory is forbidden.") The goals of universal access and thus significant impact on the educational process are not met in any way.

---

* The Student Information Processing Board (SIPB) does provide some computer access for ambitious students, but its budget gets smaller every year.

As mentioned, this approach has the advantage that it does not require any decision or innovation, and we understand pretty well its limitations. But it is both expensive and unsatisfactory.

<u>An approach that can work</u>

We have found two available systems that come reasonably close to meeting the requirements of a department educational computing facility, and we have considered two approaches to acquiring each, leading to a matrix of four possibilities. The two systems are the Digital Equipment PDP-10/PDP-20 series with the TENEX operating system and the Honeywell level 68 with the Multics operating system. The two approaches are, on the one hand, using internal financing to rent the minimum possible system, having an intent to someday expand to the scale required for full impact, and on the other hand, using a substantial gift to quickly acquire and install a system that can have an immediate impact on the educational process. For purposes of discussion, these four possibilities are numbered as in the display below.

		system	
		PDP-10-20/TENEX	HISI 68/Multics
acquisition method	internal funding	II	I
	external funding	III	IV

Being realistic, gifts of the size necessary to operate in either of boxes III or IV of this display are sufficiently unpredictable in both existence and timing that we may be forced to begin with a holding action, using either box I or II, for an indefinite period. Again being realistic, a review of the present department budget (in the previous section) reveals that we are in effect already operating in box I.

Before looking in detail at acquisition proposals, it is useful to review briefly the technical characteristics of these two systems that seem to make them technically suitable.

1) Both the PDP-10 and the Honeywell Level 68 appear to be large enough to handle the department's educational computing requirements with a single system; multiple machines and special network software is not necessary. At the same time, the standard operating system of both machines has been attached to the ARPANET, so a reservoir of network software is available for adaptation to a departmental or M.I.T.-wide data network.

2) Both systems have user-available address spaces that seem large enough for department needs in the next decade. The PDP-10 provides 1M bytes of address space, while the Level 68 provides 1024 1M byte segments. MACSYMA, for example, runs on both systems.

3) Both systems have available good implementations of LISP, APL, and FORTRAN. In both cases, a suitable PASCAL implementation would have to be imported, but both systems provide good tools for compiler importation, and PL/I on the Level 68 or SAIL or BLISS on the PDP-10 might be usable in the interim.

4) Both systems are designed to be used primarily by interactive terminals, and operate efficiently even if all their load is interactive. Neither requires a large operating staff, though Multics might benefit from some modification to its file backup strategy and initialization to make it more completely operatorless.

5) Local knowhow in both systems is widely available, among both faculty and students. This locally available knowhow means that it should be easy to find help in getting started and in diagnosing troubles.

In comparing these two systems with each other, the technical differences from the point of view of departmental use are not particularly significant.

1) Multics comes with a wider repertoire of resource management software, for adding users, monitoring usage, etc.

2) The TENEX operating system has a command language that is simpler and has fewer meaningful variations than does Multics. This simpler human interface is an advantage, pedagogically, though both systems are far from ideal in their human interfaces.

3) The PDP-10 is operated at Carnegie-Mellon, Harvard, and Stanford, among others, so a wider academically-oriented community exists with which to exchange ideas and programs. The only academic installations currently using Multics are M.I.T., the University of Southwestern Louisiana, and possibly SUNY Buffalo (via RADC).

There are, of course, dozens of other technical differences between the two systems, but none of them seem important for a department educational computing facility.

The important non-technical difference, as mentioned in the introduction, is cost. The list prices of comparable Level 68 and PDP-10 comfiguration generally differ by almost a factor of two. This difference in price, given the technical comparability described above, would normally stop the discussion, but the possibility of a hardware gift or special discount on the Honeywell equipment should not be overlooked.

A specific plan

In the light of these considerations, the following specific plan is proposed:

1) As soon as possible establish potential sources of funds or gifts of hardware for the two system choices, and choose one of the two paths. One reason for encouraging such a decision as early as possible is to concentrate present software development efforts for maximum long-term usefulness.

2) Switch to a holding action compatible with the planned system. For the case of the Level 68, continued temporary use of the IPC facility would be the simplest approach. For the case of the PDP-10, diversion of the present computing budget to rental or purchase of the smallest feasible PDP-20 configuration (costing probably around \$180K/yr., including operations) would probably be appropriate. For either case, medium speed (9.6kb) video displays should be acquired as soon as possible.

3) As soon as funds are available, purchase one of the two computer systems. In either case the system would be a department-owned computer, independent of the Information Processing Center. A full-scale PDP-10 system of the capacity needed would cost about 1.5M. A rough estimate of the corresponding Honeywell 68/80 system is \$2.5M.

If funds are acquired gradually, the configuration can be acquired gradually. (Gradual acquisition probably does not work with the Level 68, which has a very high minimum price.)

4) For the longer term plan that high performance display terminals containing, in effect, personal computers, will become the standard terminal for the department system, and that further growth in computational capacity and memory size will be accomplished by gradually increasing the capability of those personal computers. Thus the initially acquired central facility will eventually become simply a department-wide software and data library and communication repository.

We believe that this plan is reasonable, cost-effective, and responsive to the educational needs of the department. We also note with interest that Stanford University, under the leadership of Professor J. McCarthy, recently put into operation a PDP-20 system with goals and a style of operation very similar to those we have proposed.

The sketching out of this plan constitutes the last activity that this committee should undertake under its charter to recommend a course of action. If the plan is to be followed, it is essential that the department immediately place in charge a faculty member who is personally interested in leading the acquisition and installation, and also in promoting the facility as an educational tool within the department. A lot of hard work is required to create such a facility and make it a success. Without an activist leader, nothing will happen.

Jerome H. Saltzer, Chairman
Fernando J. Corbató
Michael Hammer
Frederick C. Hennie
Jin-Au Kong
Paul L. Penfield
Nils R. Sandell
Donald E. Troxel
Joseph Weizenbaum