DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

CAMBRIDGE, MASSACHUSETTS 02139

December, 1976

Position Paper II: Proposed requirements for a Department Educational Computing Facility

From: The Ad Hoc Committee on Educational Computing Resources

Overview:

This paper proposes functional requirements, estimates the kind and quantity of hardware required, and identifies the cost categories involved in acquisition of an educational computing facility for the Electrical Engineering and Computer Science Department of M.I.T. The policy position leading to a proposal for a department computing facility for education can be found in an earlier position paper from this committee, entitled "A departmental computer facility", dated October, 1976. A third paper is planned to propose one or more specific implementations based on particular manufacturers' offerings.

One purpose of this paper is to stimulate discussion on the subject and thereby assess the reasonableness of the proposed requirements. Some readers will find the proposed requirements to seem too ambitious, while others will see them as too modest. The committee has attempted to propose a reasonable balance between the possibilities of modern technology and the limitations of budgets and people's time. In particular, the requirements developed here have been tempered by several real-world considerations: the initial cost; availability of off-the-shelf hardware and software; limited resources for custom hardware and software development; and the cost of long-term maintenance of custom-developed components. But at the same time, the requirements have been driven by the goal to make a real impact on the way that engineering education is accomplished in our department by injecting universally available computation into every student and faculty member's repertoire of tools.

1) Functional Requirements

1.1) Interaction. The purpose of this system is to provide E.E.C.S. students and faculty with experience in the use of modern, interactive computation as a problem solving technique. Therefore a responsive, interactive system is an absolute requirement. Further, the primary goal is ease of use, so an easily approachable, easy-to-learn interactive command language is necessary.

1.2) Languages. There seems to be no way to make a single language serve the diverse needs of our department. On the other hand, minimizing the number of different languages used by students is important if we are to meet the objective that all students are thoroughly familiar with the facility. On this basis, we have identified needs for three languages that would be used for programming by students, with varying reasons and requirements:

A) APL. The first requirement is for a workhorse for engineering problem solving. The APL language is designed for interaction, and takes maximum advantage of it. APL is already familiar to many E.E.C.S. faculty, it is exceptionally easy to learn, and it is documented with several good reference and pedagogical texts. In addition it provides a systematic language for dealing with arrays, vectors and linear equations, so it is well-matched to many common engineering problems. The primary drawbacks of APL are its unusual character set requirements (which constrain the choice of terminal equipment), its unusual precedence rules, and a name-binding scheme that is increasingly difficult to cope with as program sizes grow. On the other hand, its unique properties are a virtue in illustration in computer science subjects that compare languages. The implementation of APL must closely adhere to the standard definition of the language, so that widely available books can be used without fuss, and it must be efficient, since it will be used extensively. Alternative choices for this language are BASIC, JOSS, and SPEAKEASY. BASIC is well-documented and does not have unusual character set requirements or unusual precedence rules, but it is pedagogically much less systematic than APL. JOSS and SPEAKEASY are sufficiently powerful, but would require a large investment to provide local support, since no manufacturer supports them.

B) PASCAL. The second requirement is for a modern algorithmic language for general programming in all areas, and for use in teaching computer science subjects, such as 6.030 (Introduction to Computation), and computer science laboratory subjects now under development. PASCAL has good pedagogical documentation available, is simpler than other languages, and provides some support for user-defined types, currently felt to be an essential concept in making programming more systematic. Its control structures are designed to permit easy construction of assertions and local proof-of-correctness. Other possibilities would be some version of ALGOL, PL/I, or FORTRAN. ALGOL has never caught on outside Europe so the combination of good quality documentation and usable implementation is hard to find, and it has no support for user-defined types. PL/I is a gigantic language, hard to learn, and filled with obscure properties and surprises for the unwary. FORTRAN is widely felt to be obsolete as a pedagogical choice, though it is essential for other reasons (see point E below). The primary problem with PASCAL is that not very many implementations are available (none yet from manufacturers) so local maintenance of a borrowed compiler or interpreter may be required. A secondary problem is that in the next few years, still "better" languages, such as CLU, may become available, supporting full user-defined type extensions.

C) The third and last requirement is for a language suitable for symbolic manipulation of irregularly structured data, to be used in teaching foundations of computer science to all E.E.C.S. students. LISP is one of the few languages that addresses this need and the local computer science community has long used it both in teaching and research. A good implementation is a requirement, and adherence to the MACLISP dialect would simplify local usefulness.

The preferred languages of use are the previous three. However, in addition to these primary languages the system must have two other languages available for "hidden" use, by faculty and staff. These are:

D) FORTRAN. The requirement is primarily for compatibility with a wide range of "canned" programs already in use in teaching departmental subjects, especially in the control area, and for access to "scientific" libraries of FORTRAN programs that calculate commonly used functions. Ability to exchange such programs with other FORTRAN users is also required. The implementation must produce good object code, and have a fairly short list of differences with the usual (e.g., IBM) FORTRAN definitions. It is absolutely essential that FORTRAN programs be usable by calling them from programs written in other languages, particularly from PASCAL and, if feasible, from APL, since it is anticipated that this will be the primary way that FORTRAN-written programs will be used by students.

E) An assembler for machine language will be necessary for some programming problems, and perhaps useful for illustration in computer science. A sophisticated macro-assembler is not necessary. Again it is anticipated that the primary use of this language will be by staff, not by students.

If other languages are available, their use would not necessarily be forbidden, although to avoid forcing students to learn too many languages, some pressure for coherence and standardization must arise from the undergraduate and graduate curriculum committees. A good implementation of BASIC would be useful to many students and faculty who are already familiar with that language. If COBOL or PL/I were available, it is likely that each would receive some modest use, either as illustrations in computer science subjects, or as vehicles for importing programs, but neither of them are significant requirements.

1.3) Other software functions needed

A) Editing/Catalog system. An easy-to-learn interactive text editor is required to prepare programs and messages. It should be embedded in an easy-to-use file catalog system that provides a separate catalog for each user, and provides for controlled sharing of program and data files among users. The file system must provide a quota of storage for each user, and provide high reliability of long-term storage. If feasible, some form of user-managed detachable storage (e.g., cassettes, floppy disks, etc.) would be desirable as a way to avoid encounters with the quota. Secure protection of one user from another is essential, since the system will be used by M.I.T. undergraduates, who are known to get their kicks by showing off their system expertise. Most APL systems include their own editor and cataloger, which may be independent. The APL system, however, must allow communication with the system file catalog, for example through APL shared variables.

B) Message system. In order that the computer system be useful as a departmental communication facility, it must have a message forwarding (mailbox) system and programs to make reading of mail easy. This system must include provision for automatic hard-copy printing and delivery of mail that is not picked up on-line. Addressing of messages should be by recipient's name; received messages must have a label that reliably indicates their source. An extension of this message system to allow user-to-user interaction would also be desirable.

C) Accounting reports. So that operational administration can be accomplished, the system must produce resource usage reports categorized by individual user and by usage class, such as subject number. (These reports would normally be directed to the system administrator, not to the individual user.) A report of the amount of usage of each terminal is also required, in order to know when locations should be adjusted.

D) Network ability. A department system must be interconnectable with other computer systems used by department members and used by people with whom department members communicate. Interconnection allows movement of programs, data, and messages among different systems in a standard way, and is important to ease of use. Connections could be expected to the department's digital laboratory network, to Delphi, to the I.P.C. Multics and IBM facilities, and to the network being devised at L.C.S. In addition, there are many small research computers used by department members; a strategy to permit their interconnection to the department facility must also be provided. Perhaps a most important property of the network ability is that it be open-ended, since the development of a network will proceed over a long period and in unpredictable ways.

E) "Front end" ability. So that regular users of other facilities do not need to maintain access to several different terminals, the network mentioned above must be equipped with protocols that allow a user of the department facility to remotely use other facilities. This ability could also be used, for example, to run the MACSYMA symbolic manipulation system for demonstrations and education experiments.

F) Resource allocation. The system must have a scheme that prevents a single user from obtaining an unfair share of computer time. We envision a multilevel scheduler that ruthlessly favors programs with short running times (less than, say, ½ second). This would be combined with a policy that limits the number of simultaneously active terminals to a number that can run short programs with good (say under 3 second) average response. An absentee/background system must be provided to allow for longer running, but non-interactive jobs.

2) Hardware Requirements

2.1) Terminals--type. Modern display technology has made it feasible to acquire terminals that are of somewhat higher quality than the terminals that are commonly used today. A communication rate of 10 kilobits per second is easily achieved on short-distance connections, and "soft-copy display" terminals (based on cathode-ray-tube/television raster scan technology) can easily take advantage of such rates. These terminals can be quiet and their speed is well suited for interactive applications. Any terminals that are to be acquired must permit use of APL, which requires both special characters and overstrikes. In addition, for other languages and message use, upper/lower case ASCII must be supported. Terminal technology is available for all of these features at a reasonable price, though no currently available terminal provides an exact match with all the requirements. Perhaps closest are the Hewlett-Packard 2641A (which operates at 4800 bps) and the Digital Equipment VT 52 (which operates at 9600 bps). Graphics capability is more difficult to firmly agree upon. The ability to display results pictorially seems to be of great pedagogical value, but is currently available off-the-shelf only in the form of (relatively expensive) storage tube displays or with special line-drawing character set options that are relatively awkward to use. Raster scan bit map display systems such as recently developed by the Artificial Intelligence Laboratory represent an interesting possibility for the future. A reasonable plan would be to acquire standard displays with line-drawing character set options now, and plan to upgrade to more sophisticated graphical display systems if they become reasonably priced commercial products a few years hence.

2.2) Terminals--number for students. Several things contribute to an estimate of the number of terminals required: some of these are known with fairly good accuracy, others only very roughly. We start by assuming that each undergraduate and graduate student averages 3 hours per week of terminal use, and that of the 168 hours in a week each terminal is available 100% of the time and is 50% utilized; thus one terminal can serve about 28 students. E.E.C.S. enrollment for 1976-7 is currently as follows:

2nd, 3rd, and 4th year department students	840
Out of department students taking department subjects	400
Graduate students	500
Number of students to be served	1740

This enrollment leads to a need for approximately 60 publicly available terminals. The biggest uncertainty in this estimate is the average number of hours per week that a student will use a terminal under a policy of universal availability. Although individual students will certainly deviate far from the average, it is hard to see how the average of student use could exceed, say, 1 hour per day, or 7 hours per week; that assumption leads to an upper bound of 130 terminals. At the other end of the spectrum, provision of only 25 terminals would be only a small step above current ad hoc facilities, which are known to be overcrowded even without a policy of universal accessibility. Finally, some reduction of load on public terminal facilities is likely if the department computer system provides for dial-up attachment of terminals located in dormitories and fraternities; some students will also use other terminals associated with research laboratories. On this basis 50 terminals are proposed for public use by students, and a project for a future Facilities Policy Committee is to monitor usage and perhaps impose 1 hour limits or sign-up sheets if a shortage develops.

2.3) Terminals--total number to be acquired. The policy of faculty involvement requires one terminal for each faculty member, and probably also one for each office containing a department secretary or teaching assistants. The number currently needed are:

faculty	117
secretarial offices	54
outlying teaching assistant offices	22
department headquarters	7
	200

There are 32 faculty, 14 secretarial offices, and 8 teaching assistant offices located in the Laboratory for Computer Science and the Artificial Intelligence Laboratory, requiring about 50 terminals; many of these already have office terminals (or nearby terminals, in the case of teaching assistants). If proper coordination with L.C.S. and the A.I. Lab. is accomplished, the department should be able to depend on those laboratories supplying their own terminals. The main problem is establishing interconnectability.

Taking into account the 50 terminals estimated for student use, it appears that acquisition of about 200 terminals will be necessary. The terminal interconnection network and computer interconnection network described below are intended, among other things, to allow attachment of previously existing terminals and thus to minimize occasions where a faculty member finds that there must be two slightly different terminals in one office.

2.4) Terminal interconnection network. The areas for which off-the-shelf solutions are most inadequate are the interconnection networks. One such network is required to interconnect 200 department terminals and terminal dial-up equipment to a computer facility. A desirable property of this network is that it allow all powered-on terminals to remain in continuous communication with the computer system, even if not actively being used. This property has three uses: it minimizes the fuss required to start using a public terminal, it reinforces a psychological feeling of being "in contact" with the system, and it allows immediate signalling that a message has arrived. The following estimates characterize the expected load on the interconnection network at the busiest hour:

terminals permanently attached	200
terminals dialed-up	25
total attached terminals	225
number actively being used	90
average number transmitting output (9600 bps)	5
average number transmitting input (100 bps)	20

In the last two items, a relatively standard pattern of "input, wait, output, think" is assumed. Since a majority of terminals will be located in one of five buildings (M.I.T. buildings 13, 35, 36, 38, and NE43) the simplest approach may be to have several local concentrators, perhaps Digital Equipment PDP-11, Honeywell level/6, or Hewlett-Packard HP-3000 computers. The concentrators would be attached to the main computer system using a computer interconnection network as described under the next point. If correctly included in the plans, this interconnection network would also allow previously purchased terminals of different types to use the department computer facility.

2.5) Computer Interconnection Network. The technology to be used to interconnect the department computer, several concentrator computers, and other computer facilities used by department members can probably best be achieved by simply imitating one of the recently constructed local networks, such as the Ethernet (Xerox Palo Alto), the Farber ring (U. of Cal. at Irvine), the Spider net (Bell Labs) or the local network planned for installation at L.C.S. and the A.I. Lab. A total network bandwidth of 1 to 3 Mb/s appears to be adequate. Some work is required to discover the number and physical location of all the currently existing and planned computer systems that should eventually be attached to this network. One of the main goals of attaching the research computers to this network is to exploit the front end ability described earlier. It may be necessary to "isolate" research computers from the network by means of buffer/concentrator computers to assure network integrity. This area has a fairly large number of unknowns, and needs more study. In particular, off-the-shelf networks aren't really available yet, so it may be necessary to add this feature later.

2.6) The computer itself. The hardest to estimate item is the size required for the computer system processor and memory, since it requires estimating both the number of simultaneous users and the average demand made by each user. If we assume that each user will consume a modest 10000 instructions per second, and we restrict the load on the computer to 90% of capacity, 90 active users will require about 1 million instructions per second (1 MIPS) delivered to them after supervisor overhead is taken out. Conservatively, then, if a centralized system is to be used, a 1.5 or 2 MIPS processor should be acquired. Assuming again an average of 40000 bytes of primary memory per active user, about 4 million bytes (4 MB) of primary memory is appropriate. This approach supposes that system effectiveness is achieved by minimizing supervisor overhead, and therefore memory multiplexing overhead should be minimized by having sufficient primary memory to avoid frequent swapping.

If we presume that each user interacts with this system about once every 30 seconds on the average, then each request will require about 300,000 instructions, or about 0.3 seconds of computation. Under the condition of 90% loading, queueing waits would add an average delay of about 10 times the service time, or 3 seconds.

Under the resource allocation scheme suggested earlier in the software function discussion, the number of users would be limited to 90; this is enough to allow all 50 of the publicly available terminals, 20 dialed-up terminals, and 20 faculty users. Students using publicly available terminals would encounter the resource limits in two ways: limited availability of public terminals (or dial-in ports), and slow response to programs taking longer than ½ second of computer time. Faculty and staff users should not normally encounter the resource limit except for occasional slow response to long-running programs. At off-peak hours, programs large in time or space requirements could run with no need for special arrangements. In general, with a fair multi-level scheduler, one would expect the available resources to be evenly divided among all requesting users.

An alternative approach, for which appropriate sizes are somewhat different, is to use a "fleet" of minicomputer time sharing systems, such as the Hewlett-Packard 3000 or Digital Equipment PDP-11. Perhaps ten such machines, each running a small time sharing system and connected in a network to allow message communication and file sharing would be adequate. Processor speed is almost certainly not the limiting factor in such a arrangement, but primary memory space probably is; because of the inflexibility in resource allocation involved in having several separate machines, one might need 6 or 8 million bytes of primary memory. Another concern, if the minicomputer approach is adopted, is the size of the largest program that can be accomodated in the address space of the computer. Although the average size of programs is expected to be small, some applications (especially canned demonstration programs) may be quite large. In particular, the 128 Kilobyte memory common on 16-bit minicomputers has been repeatedly discovered inadequate; an addressing scheme allowing programs of 1 M-byte or larger would be far preferable.

2.7) <u>On-line storage</u>. On-line disk storage is required for an interactive system of the type envisioned, so as to allow work on a project in more than one session at a terminal, or on several projects in parallel. Experience with Multics, CTSS, and ITS suggests that an allocation of an average of 400,000 bytes of disk memory per user could be adequate. With 1600 users, this leads to need for 64 M bytes of memory for users; an additional 16 M to 32 M bytes of memory should be available for libraries and expansion. These considerations suggest that one standard 200 M byte disk pack should be sufficient for a centralized system. For reliability, a second disk drive should be provided; once each day a copy of the on-line disk pack would be made and removed to a safe place. Some equivalent arrangements with the same total capacity would be needed if minicomputers were used.

2.8) Remote printers. A consequence of using "soft-copy" display terminals is that hard copy printers are also required, in small numbers, perhaps one per floor, or two in a student terminal area. Approximately 10 strategically located printers would be required in buildings 36 and 38, and five others elsewhere. Speed is not terribly important, and a 1200 bps printing terminal, such as the GE Terminet 1200 or its successors, would suffice. The only significant problem here is finding printers that can utilize both the ASCII and APL character sets.

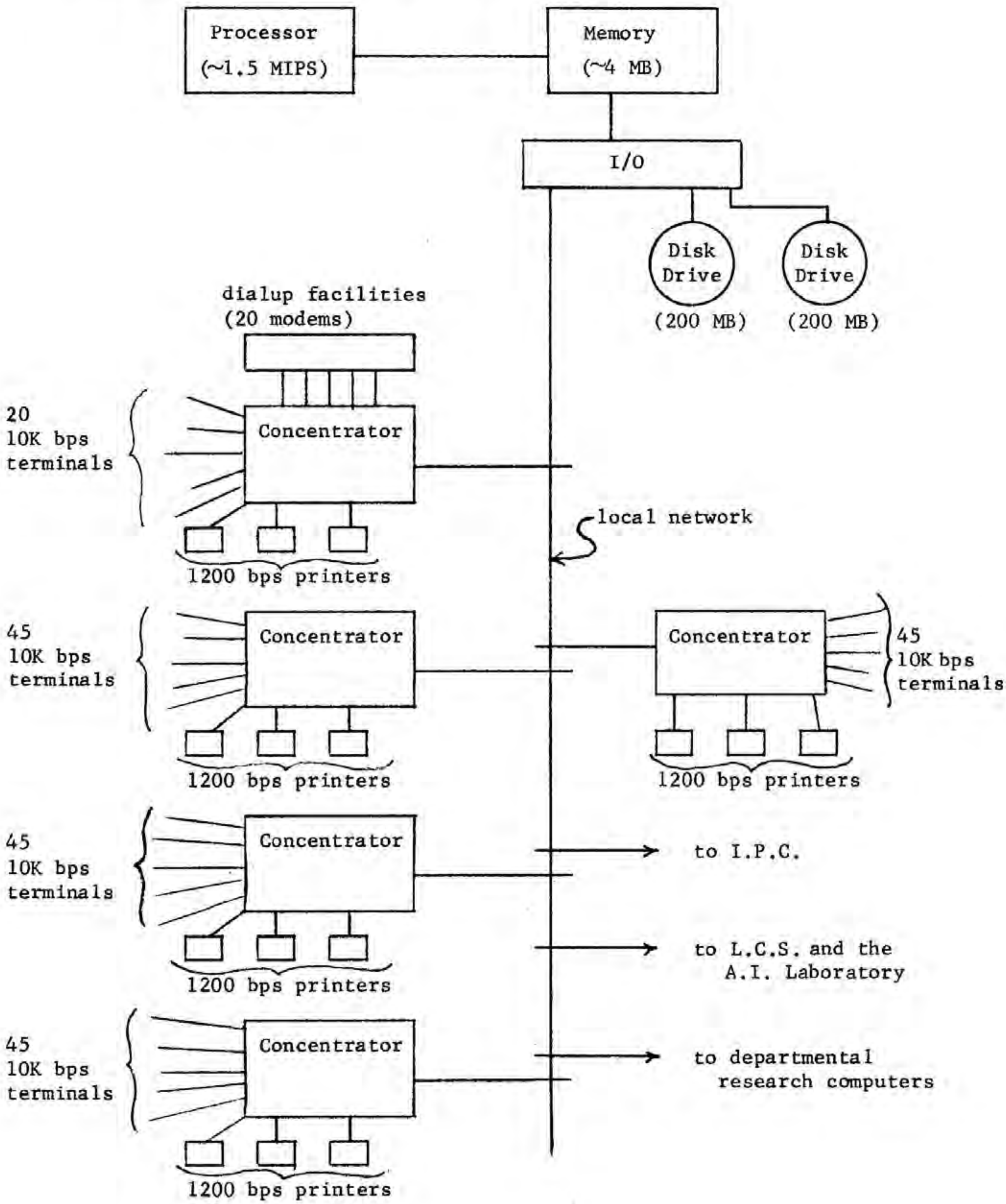
3) Stability and evolution

Stability and ability to evolve are two potentially conflicting requirements, but both are quite important in the E.E.C.S. environment. The kind of stability required is that when an instructor or teaching assistant develops a piece of software for demonstration in a class or for support of homework assignments, the programs work without maintenance when the subject is offered in subsequent years. On the other hand, the system must be capable of evolving to take advantage of emerging technology, particularly larger, cheaper memory, cheap graphical display, and large processing power in the terminal. These two requirements can probably be met by a system design that is based on the kind of terminal mentioned earlier, coupled with a plan that, when prices permit, these terminals be replaced by terminals incorporating better displays and internal computing ability. These improved terminals, attached to the original system, could absorb newly increased computational loads, while old programs can continue to run without change on the original system.

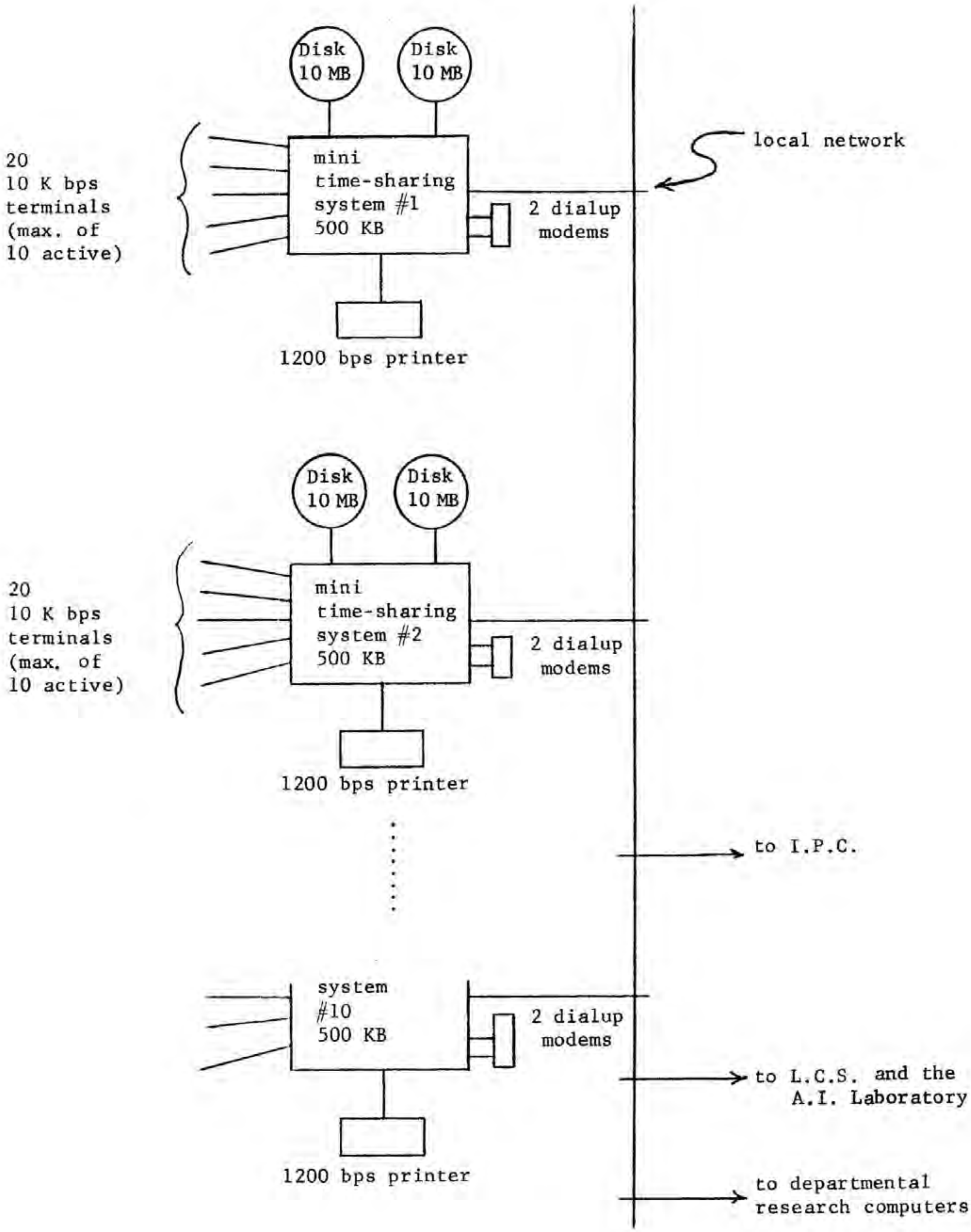
4) System Configurations

On the following two pages are shown two possible system configurations, one based on a centralized machine, the other on a decentralized "fleet" of mini-time-sharing systems. These two configurations are illustrative, not exhaustive; other arrangements (e.g., the first configuration with the central processor and memory replaced with five minis) are also feasible.

4.1) Possible Centralized System Configuration



4.2) Possible Decentralized System Configuration



5) Cost Categories

Although an accurate cost estimate can be made only in conjunction with a specific proposal naming vendors, it is possible at this stage to at least make a list of cost categories. Estimating cost is harder, since different manufacturers and different styles of operation can produce very different costs. Since some basis is needed to determine what total outlay might be involved, the following table includes three estimates, of an upper bound, a lower bound, and a "best guess". Note that the upper (lower) bound totals are not the simple sum of the upper (lower) bounds, since it is unlikely that any single proposal would achieve the upper (lower) bound in all areas. When specific configurations are later considered, the ranges should be expected to narrow substantially.

Estimating costs brings up the question of what staff are required. To help determine the cost of staffing, a set of brief job descriptions follow the cost categories. Note that implicit in these job descriptions is an image of the style of operation that is envisioned.

5.1) Initial outlay

A. Hardware

		lower bound	best guess	upper bound
1)	Processor (1.5 MIPS CPU) (or 10 mini CPUs)	\$ 240K	\$ 500K	\$ 800K
2)	Memory (4 MB) $3.6x10^7$ bits	200K	360K	800K
3)	Disk Storage and I/O channels (200 MB)	150K	200K	250K
4)	Disk packs for backup (15)	6K	6K	8K
5)	Interconnection network, including I/O channels	40K	100K	200K
6)	Concentrators with 220 terminal channels	120K	200K	220K
7)	Terminals (200)	240K	500K	640K
8)	Dial in equipment (20 modems)	8K	8K	8K
9)	Remote printers (15)	45K	75K	90K
		\$1200K	\$1950K	\$2700K

B. Installation/Operation	lower bound	best guess	upper bound
1) Space preparation/electrical/ventilating/false floor/etc. 2) Air conditioning equipment for computer room 3) Installation of cable to 200 terminals	80K	$ 90K	$ 120K
4) Custom software development	0	100K	200K
5) Manuals (200)	4K	4K	5K
	$ 84K	$ 194K	$ 325K
total initial outlay	$1300K	$2150K	$3000K

5.2) Recurring costs

A. Staff			
1) 20% assignment of one faculty member			
2) 15% of 1 administrator			
3) 1 full time programmer			
4) Student staff for operations			
5) 1 full-time technician to maintain terminals, remote printers, and modems			
	$ 50K/yr	$ 80K/yr	$100K/yr
B. Consumables			
1) Disk packs 1/month for permanent backup			
2) Paper for remote printers			
3) Documentation 50 manuals/yr for staff*			
	$ 5K/yr	$ 8K/yr	$ 16K/yr
C. Maintenance			
1) Processor/memory/disk storage and concentrator maintenance contract with manufacturer @ 10%	56K	130K	200K
2) Materials for terminal/printer maintenance	7K	10K	12K
	$ 63K/yr	$140K/yr	$212K/yr
total recurring costs	$118K/yr	$228K/yr	$328K/yr

* Students can purchase at the COOP or instrument room.

For comparison, the current identified annual cost of educational computation for the department is roughly \$100K/yr.; some unidentified costs of administration--perhaps \$25K/yr.--must also be present. These amounts represent the maximum that could be easily diverted to support a department facility. The suggestion has been made that operating costs above \$125K/yr. be capitalized, for example by including a 6-year total in the initial outlay. This approach would add nothing to the lower bound initial outlay, about \$450K to the best guess initial outlay, and about \$1200K to the upper bound initial outlay.

5.3) Staff job descriptions

A. Faculty member. (20% assignment) This person is accountable for the success of the system as an educational enterprise. The function includes setting policies to encourage the right kinds of use, soliciting ideas and encouraging faculty members to try educational innovations using the system, and coordinating uses by different faculty to avoid proliferation of languages that one student must master.

B. Administrator. (15% assignment) Enrolls users, reviews usage statistics to discover unused/overused equipment, maintains supplies of consumables, arranges for delivery of printed output, acts as referee on disk usage, arranges for telecommunications and terminal installation and movements, keeps track of equipment (especially terminal) locations. Responsible for initial installation mechanics.

C. Programmer. (full time assignment) Keeps operating system operational. Responsible for recording and supervising contents of the operating system, system-wide libraries, and subject libraries. Notifies users (instructors) of possible need to test their libraries against proposed system changes. Performs any department-wide custom software development (e.g., inportation of a PASCAL compiler) and maintenance on that custom software. (It is assumed that most of the specialized subject software will be constructed and kept up-to-date by the instructor-in-charge of the subject in question rather than by this programmer).

D. Student Staff. Provide round the clock coverage of computer room and terminal areas. Keep system up, and respond to calls about system crashes. Answer minor queries about logging in, etc. Remove and store backup disk packs. Police terminal areas. (Since some subjects with large computer use volume will need to station teaching assistants during first and second shift hours, the exact extent of need for student staff is not clear. One shift of coverage is probably a reasonable estimate.)

E. Technician. (full time assignment) Maintain terminals, remote printers, local network, and dial-up modems. Perform first look at problems with main computer system to see if maintenance call to vendor is really necessary.

5) Timetable

Since the educational objectives that this facility would meet are important and are currently unmet, the general goal is to implement the facility as soon as technically and economically feasible. The primary uncertainty is the time of availability of funds for system acquisition. If funds were available today, the following rough schedule would be possible:

1)	completion of detailed specifications, in conjunction with vendors, of three or four candidate systems	February, 1977
2)	choice of system and order placement	March, 1977
3)	installation of first equipment	Summer, 1977
4)	initial testing	Fall semester, 1977-78
5)	configuration fully operational	Spring semester, 1977-78

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