Contents:

The GEM System: A Progress Report

Edgar T. Irons

Research Report #68

May 1976

Acknowledgements

Many people in the Department of Computer Science at Yale have contributed to the GEM system. Peter Weiner, now at the Rand Corporation, directed the project for most of its duration and made technical contributions at all levels. Charles Minter developed and built the terminal hardware. Robert Tuttle managed the adaptation of the UNIX system to our hardware configuration and is responsible for integrating the graphics capability of the terminals into the software system.

The GEM System

In 1971 the Yale Department of Computer Science began a project to develop the hardware and software for a low-cost computer time-sharing system with interactive graphics terminals. Half of our goal was to provide editing and simple computing services with graphics displays for students and office personnel to use in preparing simple computer programs and documents such as technical papers and business records. The other half was to construct a system that would require a minimum of technical support and to keep the cost at a minimum. We hoped to construct a system that could be supported for about 25¢ per terminal hour of use. The project was supported by grants from the Sloan and Exxon Foundations.

Today the system is operating in production, primarily for teaching and document preparation. Sixteen graphics terminals time-share under the UNIX operating system for the DEC PDP-11/45 computer. A copy of the system is being adapted to information management for student records in the Yale College registrar's office. The University is seeking funds to acquire about ten copies of the system to use as the primary computer-access system at Yale.

Objectives

This project came largely out of our experience at Yale with more expensive systems. The primary computation facility at Yale has been a large central computer (currently an IBM 370/158) supporting batch processing and various forms of time-sharing. Our department uses a medium-scale computer system, a DEC PDP-10, for most of our instruction and research. Our experience with these systems gave us the feeling that, while for some purposes the large, versatile computer systems were essential, a considerable segment of the Yale community actually made rather trivial uses of the facilities and could be served as well on a much more modest system. We felt that time-sharing is essential and that graphics terminals would have a strong positive effect on the way we solve problems on computers.

Time-sharing was costing about $20 an hour on the large system and about half that on the PDP-10, largely because of our use patterns and environment. At these rates it is difficult to justify using computers for instruction. If a student used only six hours of terminal time a week, his tuition would barely cover it.

Another rapidly growing use of computers in our environment was for document preparation. Our PDP-10 system uses character-display TV terminals, with an editor that makes the terminal appear to be a window on a text; the user sees exactly what is in his document and can make changes merely by pointing at the spot he wants to alter and typing. Our experience has been that novices, both students and office personnel, find the system easy to learn and, as soon as they have learned, nearly indispensable. Again, however, the cost of large time-sharing systems makes it difficult to justify using them this way.

Our objective for the GEM system was therefore to provide a computer service for these functions, without the great variety of services the larger systems offer. We hoped to reduce the cost of such a service by at least a factor of ten. We felt that the mini-computer, just then coming on the market at very low cost, was just the vehicle for our GEM system. We also wanted to enhance the capability of the computers for teaching and document preparation by providing a graphics display. Several hardware developments led us to believe that we could do this too at a reasonable price. Our target was a hardware configuration costing around $100,000 supporting sixteen terminals.

Environment

Our hopes for lowering computing costs by providing the right environment were based on our experience with the PDP-10 system. That system cost about $800,000 and supports about sixteen terminals. The computer has a large internal memory (900,000 bytes), substantial disk memory (70 million bytes), and a large variety of software. The permanent disk storage is not large enough, however, to provide long-term storage for all who use the system.

The environment for the system is completely open. The system is virtually always available, and is in fact usually full from 9 in the morning to midnight. Even during the early morning hours, the system is rarely completely idle. The equipment is entirely operated by the users, and we have gone to considerable trouble to make this work easily. For example, if the printer runs out of paper, placards show how to put more paper in, and we expect one of the users to do it. Users put up their own magnetic tapes. If the system crashes, as it does a few times a week, the instructions for restarting it are posted, and one of the users does the work.

Our staff consists of an operations manager, a hardware engineer, and several student employees, together about the equivalent of one full-time person. We conserve on-line storage by providing it only for people who use the computer daily. Others keep their work on small magnetic tapes, and we give them a simple procedure for copying material between their tapes and the working storage on the disk system. To organize the use of the terminals, we provide sign-up sheets.

For a typical session on the computer, the user signs up for an hour or two at a particular terminal. At the appointed time, he copies his working information from a tape onto the disk system. After the session he copies the information he wants to keep back to tape, and the working storage is released. Thus users are responsible for managing their own information, and we can handle a very large user population with limited permanent storage.

We have used this system since 1970 for essentially all the computing activities of our ten faculty members, thirty graduate students, and thirty undergraduate majors and for teaching about 150 undergraduates each term. Most departmental reports, student theses, and such are prepared on the computer system.

The self-help environment has been most successful. Not only is the cost reduced, but the bottlenecks operators sometimes cause are eliminated, and we believe most students benefit by their exposure to the "back-room" operations of the computer. Our intention with the GEM system was to capitalize on this experience, enhancing the configuration for student computing and document preparation, stripping out infrequently used facilities, and taking advantage of the latest hardware developments for keeping down the cost. The environment for the GEM system is therefore pretty much like the one for the PDP-10: sixteen terminals clustered around a printer (with graphics capability in the terminals and the printer for the GEM). Tape units are used for casual users' permanent storage. The system is operated by its users and is always available.

Hardware

The basis for our system is the mini-computer. We chose a Digital Equipment Corporation PDP-11/45 as the main processor and developed and built our own graphics terminals. The total system could be marketed for about $160,000:

PDP-11/45 processor, two tape units, electrostatic printer	$30,000.
200,000 bytes of internal memory	40,000.
45 million bytes of disk memory	30,000
sixteen graphics terminals	60,000

The main processor is a standard-configuration minicomputer with the largest processor and maximum memory. We have experimented with a number of configurations and software systems and feel that a system of roughly this magnitude is necessary to support our objectives. The processor must be large enough to support the required amount of memory and must include memory-management facilities for time-sharing. In the DEC line, the PDP-11/40 is the smallest choice offering these facilities. Our 11/45 is only slightly larger and more expensive. With TV terminals, it is necessary to have a hard-copy device at hand. Electrostatic printers are well suited to our application, being modest in cost and capable of reproducing graphics at the same level of detail as the terminals.

The memory required to support a system that meets our objectives is fairly substantial. In all of our experience with software on the GEM system, we feel that 100,000 bytes is the minimum. Above that, additional memory has the effect of improving response time on the system. Even with the modest demands our users make, the additional memory significantly improves the system.

The disk storage on our system is fairly capacious for our original objectives. Minimum disk storage is probably about 5 million bytes. At that level, the system's response time would be degraded somewhat, and there would be no room whatsoever for storing users' data between computer sessions. At our current level of disk storage, we have enough memory to store permanent information for frequent users.

The number of terminals is arbitrary, but to minimize costs we wish to have a fairly large number on one processor. On the other hand, too many terminals gives the system erratic performance, very slow when all are working hard. Our experience with other systems suggested ten to twenty terminals for this system, and we chose sixteen.

Our major contribution to the hardware for the GEM system was the development of the graphics terminals. Graphics was available in computer terminals only at a high price, but several developments in hardware led us to believe that we could have graphics terminals at a low enough cost to make them feasible for our system. More than that, we felt that the real impact of graphics in computer systems depended on making graphics a capability of the standard terminal. We feel that the impact of our editing systems for character TV terminals has been substantial, and that the impact of the GEM graphics terminals will be even more so. We hope to make this type of terminal the standard at Yale.

We investigated a number of technologies in our search for graphics terminals. Most graphics displays up to 1970 were implemented with line-drawing systems using high-quality (and expensive) cathode-ray tubes. We felt that standard rastor-scan TV systems (with an off-the-shelf TV monitor for display) would give us acceptable quality at a reasonable price. The main problem with this approach was the substantial memory required to store the dot pattern on the TV screen. Our current system uses 32,000 bytes of memory per terminal. One system on the market at the time we began used drum memory to store the bit pattern. This system, manufactured by Data Disk, was still fairly expensive and suffered from the awkwardness of accessing and altering data on a disk memory. Another system, the PLATO terminals developed at the University of Illinois, was just becoming available. That system used "plasma panel" displays, in which the memory is in effect on the display medium. We felt this system was well suited to our objectives but, as things developed, the price of the plasma systems appeared to be too high. A third system we considered was the storage tube used by Tektronix. The cost appeared to be reasonable, but one cannot erase individual parts of the TV screen, only the whole screen at once. We felt that this characteristic would make it difficult to do graphics editing, one of our main objectives.

The possibility we finally chose was to use integrated-circuit memory to retain the dot pattern on the TV screen. Such memories are well suited to this purpose, since they must be continuously interrogated and rewritten to retain their contents anyway. Getting the display data from the memory can therefore be done as a part of this refreshing process with little additional expense. More than that, such memories can be put in as a regular part of the computer's main memory. This has two significant advantages. Altering the dot pattern is done easily and quickly merely by altering the bit pattern in the computer's memory. And the memory can be used to hold normal programs and data when it is not being used for the display.

The cost of a terminal using this display scheme is about $4,000:

32,000-byte display memory	$3,000.
standard TV monitor	300.
standard ASCII keyboard	200.
housing	300.

We believe that the potential for cost reduction on the principal component, the memory, is very good, since these memories are becoming the most widely used computer memories and the market for them is huge. More than that, because the memory can be used for computing when it is not being used for display, it is essentially free in many applications.

Unfortunately, the total amount of memory for the terminals, 500,000 bytes, is considerably more than our mini-computer can handle. We therefore built a special interface between the terminal memory and the processor, which in effect allowed the computer to consider part of the entire terminal memory as its own, selecting which part by a switching arrangement. The interfaces developed have considerable flexibility, but our standard procedure is to connect the memory of one terminal at a time to the computer. In a time-sharing environment, it works out quite well to hook up to the processor the terminal memory for the user the processor is currently serving.

More unfortunately, the memory we purchased appears not to be reliable enough for storing programs and data. An occasional missing bit does not matter much for a display, but often matters a great deal to a computer program. In addition the software system we finally adopted is not prepared to use the memory of our terminals for storage, so we have not been able to take advantage of the potential dual use for the terminal memory. But this shortcoming is just an unfortunate outcome of our implementation. There is little question that such memories can be made reliable enough for regular computer memories, since many computers now use them for main memory. Nor are there any fundamental complications in having an operating system use the terminal memory for both display and programs, since we developed and tested a system that did so.

We had hoped that these ideas would prove attractive enough so that some company would pick them up and market the terminals, but so far this has not happened. Intel has considered marketing the memory boards with the TV signal output already built in (this involves hardly any additional manufacturing cost). MIT has privately undertaken to implement systems based on our ideas, but they are not commercially available.

Software

We wanted the GEM system to edit in the fashion of the Yale Editor [1,2] eventually extended to pictures as well as text. We felt that some simple computational capability should also be included. At the time when we began our project, none of the operating systems available for the PDP-11 was at all adequate for our goals, so we undertook to develop the entire system on our own.

We chose the APL language for expressing computations in the system and designed an implementation to integrate our TV editing concepts with APL and graphics capability. We have described the system elsewhere [3], so we shall not give details here. The system was developed to the point where it was fully operational (in fact it was used in Yale courses for the fall term of 1974), but the manpower to harden it into a solid production system and to maintain it was not available.

At the same time, the UNIX system, developed at Bell Telephone Laboratories for the PDP-11, was being made available free to universities ($20,000 to others) and we found it to be an excellent operating system, fully developed and in production use at Bell, and supported by them. On balance we felt that the UNIX system was a better basis for development and adapted it to our terminal system. This system has been in production use at Yale since September of 1975 and has been well received. We were fortunate to be able to get a TV editing system based on the Yale Editor that the Rand Corporation developed for UNIX. Our computational capability is obtained through FORTRAN, BASIC, and the Bell language C.

The adaptation of UNIX for the GEM system has forced us to reconsider our APL-graphics-editing combination, but many of the effects we were trying to achieve are available in another form, and it is clear that we could develop our system with UNIX as host. Currently, we do not have the manpower or the funding for this undertaking.

In a side development, the original GEM software has been adapted for managing student records in the Yale College registrar's office. This system is just now being put into operation, so it is too early to evaluate the results, but we have high hopes for our approach. In fact, we suspect that a system like the GEM may become the model for office management at Yale and elsewhere.

Networks

While we based our GEM system design on the proposition that most computer users don't need any more facility than the GEM provides, some users certainly do. We felt that we could serve this audience too by providing them a means for sending programs to a larger computer for execution. For this class of people, the GEM would be an editing machine and a terminal for larger computers located on the Yale campus, or even in other cities. Even for people whose computational needs the GEM can satisfy, the ability to exchange data with other computers or other GEMs would clearly be an advantage. For example, in the student records system at Yale the objective is to keep records locally in the registrar's office. But other computer systems need access to the information for other purposes, so the GEMs must be able to communicate with other machines.

While we have worked with several communications links, our goal of a well-developed network facility remains largely experimental today. We have connected our Computer Science computer complex to the main Yale Computer Center machine and to the AEC machine at New York University. The NYU link gave Yale physicists ready access to a machine they were obliged contractually to use for their work; they had previously had to travel to New York. This link was finally put into production, but is itself still isolated from most other Yale machines. The link between the Computer Science machines and the main Computer Center is still somewhat awkward and is not widely used. Even links between the GEM system and the PDP-10 are not yet satisfactory. Thus, while we have explored a number of network ideas, realistic and simple network connections have not been completed. The work continues, and we still believe this interconnection is important. It will be even more important if the GEM system is replicated a number of times at Yale. Again, inadequate funding makes our progress slow at the moment.

Costs

An important objective for our system was low cost. We believe we have succeeded in this regard, but two factors make it difficult to judge exactly what the costs of running the system are. One factor is that our terminals were developed here in our laboratory and are still not commercially available.

The other is that our operation within the University does not charge out building space and environmental costs for the GEM system or even for the department, so we can only estimate these expenses.

We estimate the hardware costs at $160,000. $100,000 of this is the current purchase prive for the off-the-shelf equipment -- everything but the terminals. It would take us about $60,000 to reproduce the terminals we are now using if we did it ourselves in Yale's electronics shop. Terminals of this kind are not commercially available; if they were, they would probably be somewhat more expensive. Graphics terminals with less capability (the Techtronix terminals, for example) can be obtained for about this cost. TV terminals without graphics capability can be purchased for about half this cost.

Amortizing the hardware costs over five years with interest gives us a yearly cost of about $40,000; other costs for running the system, maintenance, supplies, and supervision, should run about $30,000 a year. We estimate maintenance at 10% of purchase price a year, a fairly standard practice; supplies at $5,000; and supervision at $9,000. Supervision includes part of a manager, part of a secretary, but, consistent with our philosophy of operation, no operators or development costs. Our experience supports this estimate, since it costs us about $60,000 a year to operate our two computers, the GEM system and the PDP-10, and the 10 is considerably larger and more complex than the GEM.

The total yearly costs, then, are about $70,000. The cost of an hour at a terminal on the system is found by dividing the yearly cost by the number of hours the system is in use. The minimum cost would result from total saturation, 24 hours a day, 365 days a year. Saturation during normal office hours would give forty hours, fifty weeks a year. Saturation in an academic environment would be greater during session, since students also work evenings and weekends, but less when school is out; we estimate academic saturation at fifteen hours a day, seven days a week for the thirty weeks of classes. So we can get lower bounds on the costs in the three environments:

Total saturation

16 terminals × 24 hours a day × 365 days a year = 140,000 hours

$70,000 yearly cost ÷ 140,000 hours = 50¢ an hour

Office hours saturation

16 terminals × 40 hours a week × 50 weeks a year = 32,000 hours

$70,000 yearly cost ÷ 32,000 hours = $2.20 an hour

Academic saturation

16 terminals × 15 hours a day × 7 days a week × 30 weeks a year = 50,000 hours

$70,000 yearly cost ÷ 50,000 hours = $1.40 an hour

Actual use will of course be below these figures, probably by a factor of two. Based on our experience in the first year of operation of the system, we expect about 10,300 hours a term for two terms plus some light use in the summer term. Thus the actual use is around 25,000 terminal hours, or just about half the maximum level.

Actual Yale use

$70,000 yearly cost ÷ 25,000 hours = $2.80 an hour

Another measure of system cost in an academic environment is the cost per student. The maximum use we would expect a student to make of the system for one course is six hours a week. Running at this level seven days a week, fifteen hours a day gives us the capacity of at least 280 students at any one time. Over two terms we serve 560 students a year, for a cost of $125 per student per course. We actually serve about 600 students a year because some courses require less than six hours a week on the system.

Yet another way of viewing the cost of the system is to compare it with stand-alone machines. Each terminal in effect costs $10,000 to buy and $104 a month to maintain. This figure is almost the exact cost of a small stand-alone computer system like the IBM 5100. To achieve the same costs on a $2 million medium-scale computer would require that the machine service 200 terminals. We believe that in the current marketplace the GEM system compares favorably with either of these alternatives. Some advantage is gained in capability over small stand-alone systems by clustering sixteen terminals around a shared processor. Limiting facilities and distributing the system produces savings over large systems.

References

1. Edgar T. Irons and Frans M. Djorup.
   A CRT editing system.
   Communications of the ACM 15:16-20, 1972.

2. Edgar T. Irons.
   The Yale Editor.
   To appear as a Yale Computer Science Research Report, 1976.

3. Edgar T. Irons.
   Software for a graphics editing machine.
   To appear as a Yale Computer Science Research Report, 1976.

References

[1] E. G. Belaga.
Some problems involved in the computation of polynomials.
Dokl. Akad. Nauk 123:775-777, 1958.

[2] R. V. Book and R. J. Lipton.
Unpublished manuscript.

[3] R. J. Lipton and D. P. Dobkin.
Complexity measures and hierarchies for the evaluation of integers, polynomials, and N-linear forms.
In the proceedings of the Seventh Annual ACM Symposium on Theory of Computing, 1975.

[4] T. S. Motzkin.
Evaluation of polynomials and evaluation of rational functions.
Bull. Amer. Math. Soc. 61:163, 1955.

[5] M. Paterson and L. Stockmeyer.
On the number of nonscalar multiplications necessary to evaluate polynomials.
SIAM J. Computing 2:60-66, 1973.

[6] V. Strassen.
Polynomials with rational coefficients which are hard to compute.
SIAM J. Computing 3:128-148, 1974.

[7] S. Winograd.
On the number of multiplications necessary to compute certain functions.
Comm. Pure Appl. Math. 23:165-179, 1970.