Jamie Blask 2014.09.23 09:13:40 -04'00'

A Response to the Technologies Framework Draft 0.2:

The 'Tao of Computing' or a 'Network to Nowhere'?

Preamble

I believe the goals of CIT should be:

1) to facilitate *global* communication among students, scholars, staff, and other parties, while ensuring security of confidential data;
2) to support administrative computing needs;
3) to support advanced research;
4) to support key information technologies at competitive prices;
5) to put the needs and desires of users first.

Some means for achieving these goals are:

1) providing a secure mail interface for users of all popular systems, providing access to local and national bulletin board systems, promulgating standard data formats for file exchange, and providing network connectivity extending to anyone with a telephone line;
2) developing a flexible computing and database infrastructure allowing users to share administrative information and distribute information processing tasks to departmental servers or workstations;
3) providing a research infrastructure offering state-of-the-art computing and network facilities for those requiring access to high performance technologies;
4) providing cheap access to popular mainframe environments (VM/CMS, VMS, and UNIX), providing highly competitive sales and service for popular information technologies, and promoting open technology standards which are capable of evolving to higher performance/price levels as fundamental hardware and software technologies advance;
5) conducting surveys to determine what users want, and doing our best to either satisfy their desires or explain why they cannot or should not be satisfied. This service should be provided through CUINFO.

Numerous pitfalls exist along this path. To list a few which are related to technology assessments:

Digital Centralism: The belief that centralization of resources is inherently superior in price and performance, and that users should applaud the pre-emption of resources by slow, expensive central service bureaus. This belief has been regnant in MIS centers across the country... which are now being shrunk or dismantled at an accelerating pace as users wrest control of their destiny from these erstwhile experts.

Digital Anarchism: The belief that everything will work out best if users do exactly what they wish. As Digital Centralism has crumbled under the pounding of inexpensive microcomputer technologies, Digital Anarchism has waxed, with dismal consequences for large organizations as communications systems fail to interoperate and departmental data becomes isolated in computing archipelagoes.

Performance Pathology: The belief that maximum performance is the most important criterion for judging technologies. This could be the floating point

performance of a computer, the bandwidth of a network, or the "richness" of a user interface. A fixation on performance in technology assessments becomes pathological when it is combined with *Egocentrism*, revealed in the assumption that those who lack the resources to acquire the latest computing goodies are beneath consideration.

Vendorism: The belief that some vendor possesses unique vision and competence which should exclude other vendors and their technologies from serious consideration, even as the vendor strives to lock its customers into expensive proprietary solutions. This condition is displayed in its most pitiful form as an attachment to "traditional" technologies in their declining years, even as they are inevitably surpassed by new technologies which offer not only fresh approaches to problems already addressed, but solutions to problems which had not been considered soluble. A phenomenon known as "Party-Lining" reinforces these tendencies: Vendorists attend conferences of their cohorts, where they are lavishly treated by the One True Vendor, and through the calculated application of the *psychological* technology of cognitive dissonance they are lured into the fallacy of...

Futurism: The belief that some soon-to-be-released technology is the only appropriate solution to current problems (including problems which can be addressed with current technology). Generally, the details of this technology are secret, the technology is and will forever remain proprietary, and even vague information is released only after a non-disclosure agreement has been signed. Futurists invariably compare this amazing technology of a shining tomorrow with the present-day condition of competing technologies, and disdain birds roosting on the hand—the better to pursue a Phoenix on a mountaintop.

Digital Transcendentalism: The belief that information technologies can resolve every problem (or, alternatively, the less extreme belief that problems which do not submit to technological solutions are not worth considering within a broader whole-systems framework). The Digital Transcendentalist often envisions a future of unrestricted access to a communal Hyper-Reality, with users remotely processing shared data transparently, in an open environment of trust and confidence—something like the Free Love gospel of the '60s in a digital environment. No one will be so rude as to disrupt this placid environment with worms, viruses, untranslatable data formats or incompatible interfaces, or completely unforeseen technological breakthroughs rendering the present state-of-the-art obsolete. Unfortunately, everyone intimately involved in information technologies suffers to some degree from this condition.

These are just a few of the pitfalls; every foible of human reasoning can be mis-applied in this complex, fast-evolving field.

My Reaction to the Draft Report

I believe the Draft Report suffers from some of the fallacies listed above, particularly where it deals with server and workstation technologies. For example, it exhibits a tendency, in its rather extreme statements of support for "distributed computing" technologies, to espouse a Futurist Great Leap Forward from our present state-of-the-art-ca.1985 environment to a 1995 environment; both the realities of the 1990 computing world and binding economic constraints tend to go unexamined.

The draft suffers also from a lack of editorial attention. It compares poorly with other documents which have been prepared for CIT guidance. There are even contradictions in the report—sometimes within the same sentence! Before I try to outline my perceptions on the technologies issues, I will first address what I perceive as the substantive flaws in the report.

Some Errors of Fact...

> "An institutional file server will provide transparent access to files across a wide-area network at speeds comparable to local disks." (p. 5)

The physical laws which govern time and space govern communication, too; the speed of information transmission will forever be limited by some kind of inverse-square relationship: as distance increases, the speed of communication will drop, and not proportionately, but dramatically. Observe also that serving multiple users rather than a single user adds communication overhead, and it is evident that remote file service will probably always be significantly slower than local storage.

The report also states that "In general, local storage costs more but performs better with higher availability than a remote server supplying data across a network" (p. 14), which would seem to contradict the previous statement.

> "As network speeds reach 10Mb/sec service to the desktop, shared file servers become a very attractive alternative to hard disks in the workstation." (p. 19)

This is true for the case of a file server which is located on a subnet with bandwidth to spare, for users who need to share files frequently or share access to databases. A fileserver can be much more convenient, but it is generally not cheaper than local disks.

> "... locally owned hardware to store data only seems much cheaper than rates charged on mainframes until the fully loaded costs of backups and archive/restore procedures are factored in." (p. 27)

Maybe—if you expect users to buy machine rooms, air-conditioning, and full-time staff for systems which do not require them. At the current price (~$14,000) for storing 100MB of data for *one year* on one of Cornell's mainframes—and this does not include the similarly steep cost of *accessing* that data—you can purchase, for example, a NeXT machine including database software, a 256MB optical disk for backup and archival, an additional 330MB internal drive and a laser printer, and *still* have thousands of dollars left over to cover administrative overhead. For the cost of mainframe data storage *alone* you can buy a system which will offer not only superior response times, but the ability to chart your data, analyze it statistically, integrate it into the text of your research paper, format the paper, and print it over and over until it looks just right.

> "Workstation operating systems will ... have limited capability to utilize co-processors." (p.6)

To the contrary, most microcomputers today can support a math co-processor, many employ I/O coprocessors, and relatively cheap microprocessor-based platforms are becoming available which can use multiple CPUs to perform symmetric multiprocessing, boosting "microcomputer" performance into what was once the "superminicomputer" domain; serial I/O boards are available which use an 80186 and up to a megabyte of memory to offload burdensome serial I/O overhead from the CPU; caching disk controllers can reduce hard disk average access times to ~.5 msec; the DSP (Digital Signal Processing) chip in the NeXT machine is arguably more powerful than the CPU.

Modern bus architectures for microprocessor based systems, such as the NuBus, the Micro Channel, and EISA, support multiprocessing configurations. Many recently introduced microprocessor CPUs support multiprocessing, e.g., the Intel 80486 and i860 and the Motorola 88000. In the software area, Mach, the portable OS kernel which is used on the NeXT machine and is being promoted as the standard UNIX kernel by OSF, was designed to support multiprocessing; Intel is going to release versions of UNIX for the 80486 and the i860 which will support multiprocessing; Compaq supports multiprocessing on its latest platform.

The truth is that the microprocessor revolution threatens not just minicomputer vendors; it threatens "traditional" mainframe and supercomputer architectures (as Omri Serlin argues in his column in the February 1990 issue of UNIX/World, "Are Killer Micros Taking Over Supercomputing?")

> "With UNIX-based systems, the operating system can also be booted from a server, but a fast local 'cache' hard disk is still needed." (p. 20)

This is an invalid generalization from the experience garnered from operating the NeXT lab (a somewhat atypical computing environment dominated by advanced computer-weenies). The reason the NeXT machine now incorporates a dedicated paging device is not because of any inherent reliance by UNIX on a local hard disk, but because the applications being run are memory hogs! Note that the NeXT is bundled with software systems (such as Mathematica and Common LISP) which often require *very* large amounts of memory, and thus rely extensively on the machine's virtual memory services. If you want to run such applications diskless without losing performance (say on a Sun), one alternative strategy would be to install more RAM in the machines, minimizing the excessive burden virtual memory paging imposes both on the network and the server.

> "... less dramatically, workstations have started to support more 'personal' computing with the increasing use of Graphical User Interfaces (GUIs) that make the system more accessible and with the increasing number of 'personal productivity' applications available on these platforms." (p. 25)

Implying that there is a microcomputer more dramatic than the NeXT machine, InfoWorld's Computer of the Year for 1989? Sun, Apollo, and a host of other UNIX vendors have *traditionally* supported too many *different* GUIs for *too damned long!* The lack of applications for workstations is due to incompatible innovation in a highly competitive market, not a lack of vision. Unlike the PC market, no vendor was powerful enough to establish a *de facto* standard, and competition has tended to predominate over cooperation even when vendors would benefit from standards. (Please note also that hardware vendors offering proprietary operating systems, including those now taking the UNIX market seriously as a result of federally-mandated standards, have in the past tended to float UNIX offerings as a means of roiling the waters rather than hooking contracts....)

> "The future computing environment at Cornell will demand that a desktop system be able to join the 'workgroup' world as well as serving 'personal' needs." (p. 25)

> "[Applications will require] the active participation of the desktop system, be it only for the local generation of the user interface (i.e., the X Windows model)." (p. 26)

Conceptually, an X server differs from an IBM 3274/3278, DEC VT100, or Tektronix 4014 *only* in the richness of its communications and display protocols. Under X Windows the GUI is managed on the *client* end, that is, on a *host* computer running the application (of course, the X client and the X server may be running on the same workstation as different processes). For some applications (e.g., bulk data entry) a VDT connected to a host by a serial port is and will remain a cost-effective, user-friendly solution—because in data entry, *keyboard and display speed* and *display legibility* are the most important qualities. For those who require no local processing capability but occasionally need access to sophisticated workgroup applications, a "dumb" X terminal connected via Ethernet may be the best solution. The moral of the distributed computing story is that the desktop should have as much power as it *needs*.

> "Currently, as noted, the workstation platforms are used mainly to support scientific computing in some specific academic departments." (p. 25)

These workstation platforms are used not only for computational tasks, but for mail, file transfer, and desk top publishing functions. Many theses at Cornell are produced using TEX on a workstation. There is nothing distinctively "scientific" about these functions, even if those in scientific and engineering fields do tend to be more sophisticated in their uses of computers and to demand more capable platforms.

"IBM ... Apple ... the two micro platforms." (p. 26)

This is vendorism at its worst. These two vendors are indeed market leaders; however, they do not dominate the market, and many other technically innovative, financially sound vendors possess a substantial share of the market. Here is the current market share of the top 15 vendors in the global microcomputer market (from Datamation, June 15, 1989; data were provided by The Yankee Group):

Company	Market Share (%)
IBM	25.5
Apple	10.5
Compaq	7.4
Olivetti	5.1
Tandy	4.4
Zenith	4.4
NEC	4.3
Toshiba	3.9
Unisys	3.7
Amstrad	2.8
Hewlett-Packard	2.4
AT&T	2.2
Commodore	2.1
Fujitsu	2.1
Matsushita	2.1
	82.9

"Entry-level RISC-based CPUs in personal workstations are probably two years in the future... RISC already plays an important role in UNIX-based platforms..." (p. 26)

That is, a workstation running UNIX is inherently not a *personal workstation.* This notion is just plain stupid, and is unfortunately typical of the anti-UNIX prejudices displayed in the draft document.

"MacOS System 7.0 and later will continue to provide a proprietary environment that is significantly advanced in providing functions not available in other systems." (p. 39)

"Will continue to provide." This error in tense reveals a profound bias. The Macintosh remains and will remain *behind* in providing advanced OS functionality (such as memory protection, interprocess communications, support for transactions processing, support for multiprocessing, etc.) even after 7.0 is released. *Someday* System 7.0 will be *available*, or at least *most* of it will be available, and it won't be *much* more than six months late. Until then we should restrain our appetites; the repast which Apple will provend will be in every way more delectable than anything already on the table. With *John Sculley* wielding the knife and blending the sauce, our computing tastes will be blandished to a whole new level of—simplicity. Here, have another Pepsi! (:-< Please refer to my companion piece, *Ten Macintosh Terrors*, for my assessment of the Macintosh's "advanced functionality.")

"As the use of information technologies becomes integrated into the work environment, the current installed base of obsolete machines will move off into isolated corners to sit next to analog telephones." (p. 39)

Not a chance; those machines will be snapped up as *bargains* by the less rich of us. Anyone who has hassled with Cornell's shiny new digital phones will understand why: sometimes it pays to be in the following wedge of technology. (Someday, that digital phone system is going to pay off big. Someday... is *mañana*, when the local phone companies get around to implementing ISDN. "Darn! I can't figure this blast-be-doggled phone system out! We shoulda bought that system from AT&T!" *Twilight Zone Theme swells...* "Boss—we *did!*" *Fade as parties gape and exchange horrified gaze.)*

Misadventures Advocated...

"traditionally ... traditionally ... traditionally ..." (passim)

If you want a *traditional* computer, I would recommend an abacus. If you want a *modern* computer, I would recommend that you *jettison tradition. To wit:*

"Large numbers of academic departments have their own processors ... with the most popular configuration being a VAX..." (p. 21)

"The RT follow-on [the RS/6000] ... will compete favorably against the VAX as a departmental machine." (p.24)

"The line of VAX processors has been a favorite ... for years ..... VAXen are well positioned as providers of midrange network services..." (p. 24)

"The DEC VAX has been a traditional favorite on campus. ... Increased support and experience ... [with] the VAX platform will be a *requirement* for a number of services we will want to provide." (p. 38, italics mine)

In my opinion, VAXen are indeed well-positioned for an anchoring role in 1995. Anchoring *boats*—not *networks!* The RS/6000 will *not* "compete favorably" with the VAX: *It will eat its lunch!* RISC-oriented architectures enjoy a considerable performance edge; in the future other innovative architectures will doubtless supersede RISC. (Gordon Bell, the chief architect on the VAX, today heads Encore and builds multiprocessing superminis based on commodity microprocessors running UNIX over a Mach kernel. Gordon Bell on RISC, UNIX Review, February *1986*: "...if you ignore all the RISC proselytizing and focus on what is really important in that work, you'll see that it *is* the right way to build a computer now.")

Combine this performance lag with a proprietary CPU/OS comination and a closed, proprietary bus architecture (VAXBI) and what you have is an architecture that only an addict could love. VMS is a well-regarded operating system, but the abandonment of Digital's VMS-on-RISC project is clearly the "writing on the wall" for VMS users, who need to be prepared for a future in which they will lag behind the pack in terms of computational performance. Of course, as long as there are diehard VMS users, and there is worthwhile software available only in this environment, VAX/VMS will need to be supported; but giving it a crucial role is (to be polite about it) misguided.

"Character data will give way to image, video, and sound information." (p. 5)

We will all, presumably, be illiterate. The truth is that a million images of lifeless bodies will never convey the information contained in the word "genocide." It's going to be some time before technologies will be available to scan images in the same fashion that regular expressions can scan text (e.g., "I want to see all the pictures in this image database which suggest violent death due to interracial conflict between Caucasians and American aborigines...").

"Timesharing on a host computer or 'personal' computing on an isolated PC will be replaced by a distributed model of computing..." (p. 6)

"Timesharing use of VM/CMS, VMS, and UNIX systems will all but disappear to be replaced by processes on large machines that provide resources and information to processes running on workstations." (p. 6)

The operative words here are *"replace"* and *"all but disappear;"* the operative words should be *"supplemented"* and *"diminish."* Timesharing with a host using an X Window terminal remains time-

sharing, despite the sophistication of the interface technology. Timesharing will *always* remain the most cost-effective way to serve users who require access to powerful CPUs on an occasional basis.

> "The character-mode interface ... will be replaced by a graphics interface that is reasonably standard both for the workstation user and for the applications developer." (p. 6)

Once again, this verb, "replace." This fallacy follows on the heels of the assumption made above that we all would prefer to be illiterate, and will be incapable or indesirous of dealing with textual interfaces, even interfaces using menus or intelligent prompting. We will all be chained to mice, even though the use of character-based interfaces can enable the construction of highly portable systems with faster response times. To provide a friendly user interface, the Application layer can use a standardized character-oriented interface on the back end while presenting a friendlier face to the user via the GUI. (The PostScript language provides a very successful example of a system influenced by the UNIX byte-stream I/O paradigm; designing the system interface as a data processing language insulates the higher level system from low-level data type compatibility problems such as byte sex, and enhances portability; indeed, it enhances *clonability*, as the case of PostScript shows. The Page Description Language to be used in Apple and MicroSoft's TrueType system is an "extended" PostScript clone.)

I believe that a network application offering text services should offer a character-based interface even when the text is intermingled with other data types; otherwise it must be considered improperly layered, since a higher-level interface is being required to return data which could be handled at a much lower level. Requiring a fancy GUI for text services disenfranchises people who use the "wrong" kind of system—and given the variety of systems available, they will probably constitute a majority.

Graphical interfaces work well for lowering the learning curve, and are uniquely appropriate *to graphical applications;* graphical interfaces to *character-based* applications can be inappropriate, and are downright *dysfunctional* when implemented in a fashion that does not provide for automation of tasks using scripts (the lack of a scripting language for the Macintosh is widely held to be a significant drawback even by proponents of the Mac; Apple promises to remedy the lack in System 7.0). The problem here is that the semantic content of a mouse-stroke or button-down is *nil* unless you record the whole state of the user interface, a tedious and expensive chore.

And to further complicate the issue, some people find that the Macintosh desktop interface is *cluttered* and *counter*-intuitive! Anyone who believes that icons should supplant characters should consider the language of the Chinese, which *is* ideographic and which *does* require a lifetime to master.

> "The new NeXT computer, with its advanced technology and its use of a sophisticated GUI running over UNIX, is a machine that in many respects looks ahead to such a future. It comes closer than other current machines to combining the microcomputer and the workstation [?]. While this particular platform may not be the solution for The Cornell [sic] needs, it demonstrates how many of our needs in the next three to five years may be met." (p. 27)

I will yield momentarily to Our Lady of the Computer Church: "... no doubt by some other vendor, now who could it be... Compaq? Nooo... Commodore? Mmmmm... or perhaps... could it be... *Apple*?" CIT's Technologies Committee: " Well, we *could* buy it right now at a competitive price, but it ain't available in red, an' it ain't available in blue, an' we don't want it."

> "... the basic processors used ... in the 1980s will not be sufficient in the 1990s." (p. 6)
> "The Macintosh's use of the 68000 will not survive much longer." (p. 26)

The "Technologies Timeline" lumps together the 8088/8086, a chip which was BDOA ("Brain-Dead-On-Arrival") when it was incorporated in the IBM PC in 1981, and the 68000, the product which made Motorola the largest vendor of CPUs for sophisticated workstations. The much-heralded Apple Inc. still foists machines based on primitive 1978-era 6502 architecture on users; why should we expect the chip employed in Apple's shiny-new portable Macintosh to fade away in 1990?

Futurists tend to ignore that cloud lowering over our whole existence, what econonomists call "the budget constraint," and assume that users have infinite funds to apply to infinitely sophisticated systems. 6502 and 8088 systems, alas, still sell, because you *can* do useful work with them and they cost a couple of hundred dollars less than the more powerful alternatives. *Lucky* Cornell students will be bringing such systems along with them long past 1995. The *unlucky* student (say, a graduate student from Sub-Saharan Africa) will *still* have *no personal computer at all.*

> "Several factors will flatten out the rise in local hard disk requirements ... Bernoulli-box technologies, with speeds that already compare favorably with traditional hard disks, can be used to satisfy local storage requirements in the meantime." (p. 20)

The Bernoulli box is a nifty but proprietary technology, which offers reliable data storage and archival—at an entry price 4 times that of a hard disk. Bernoulli boxes are also rather large.

> "Transparent data access is a central requirement, not only on data storage systems, but also for operating systems and network interfaces. The task of getting all the interfaces working together is huge and the opportunity to not accomplish anything because not all the problems can be solved is very great." (p. 37)

No kidding! If transparent access is a *central requirement*, homogeneity of the hardware and software base had better be required first. To cite one observer (Jim Gray, "Transparency in Its Place," UNIX Review Vol. 5 No. 5): "The lunatic fringe of the distributed database community promises transparent access to heterogeneous nodes ... meaning, presumably, that all the nasties of networking, security, performance, and sematics can be hidden under a veil of transparency. Although this sounds like a wonderful scenario, the prospect of getting concurrence on a means of data-sharing from people who can't even agree on a way to represent the letter 'A' seems a bit far-fetched." Not to mention formats for arithmetic, files, sounds, and images...

> "The three leading user interfaces will converge to basically the same thing with vendors struggling to differentiate their product while maintaining that it works the same as every other product, just better." (p. 39)

Humbug. Who really believes that any of the "three leading user interfaces" are still going to be around, come the millenium, when we put aside our keyboards and mice in favor of a super-duper look-Ma-no-hands HyperReality interface?

Some Contradictions...

> "An increasing number of researchers are making use of high-end workstations (e.g. Sun or Apollo)..." (p. 21)
> "Responses to a Summer '89 departmental survey showed: Workstation is [3270, 15%, PC 46%, Macintosh 39%]" (p. 21)

A 3278 does not a workstation make; moreover, the above figures add to 100%, contradicting the immediately preceding statement about workstation use.

> "An institutional file server will ... [function] at speeds comparable to local disks." (p. 5)
> "... local storage ... performs better than a remote server" (p. 14)
> "Current networking software involves the server in significant amounts of the protocol processing[,] which becomes a very severe problem as network speeds rise by several orders of magnitude." (p. 9)

The first statement (in the Executive Summary) is not in accordance with the other statements.

Some Omissions...

I will list but a few...

UNIX support. Support is promised for the MacOS (which one?) and OS/2. Support for at least *some* flavors of UNIX should be promised, but is not. I just can't figure out how CIT is supposed to support "UNIXification" without supporting *UNIX!*

By providing plug-and-play application binary compatibility over networks of vendor-variegated platforms spanning a broad range of performance, standard UNIX ABIs will broaden the market for shrink-wrapped business applications. Take for example SCO's Open Desktop, incorporating the X.desktop window manager, X Windows, UNIX System V Release 3.0, multiple virtual DOS machines, TCP/IP, and an SQL database; it can run not only any pre-3.0 UNIX SV-compatible binary and any XENIX-compatible binary, but almost all DOS binaries. This is a very attractive environment for advanced applications; many of the most popular DOS packages are now available to run under UNIX, and many more will become available in the future.

NextStep and similar packages based on C++ or Objective C have great potential to reduce the costs of program development while supporting a multi-vendor environment. The draft mentions CASE and the possibility that some "one program fits all" development environments may become available, but doesn't adequately address the subject of future software development tools and environments.

Apple's A/UX is not mentioned, even though version 2.0 promises to fix most of the problems with earlier versions, including lack of MultiFinder compatibility. Apple's UNIX is inexpensive and works. With the advent of the IIfx, Apple will *finally* offer a machine suited to running UNIX.

The costs of different alternatives are not analyzed; and no strategies for maximizing return on Cornell's investment in IT are proposed. I believe Cornell should adopt a strategy emphasizing open standards, and shun proprietary technologies whenever possible. If corporations want Cornell to adopt such technologies, *they* should pay for the privilege.

Viral and similar hazards to computer users go without mention. I get a little nervous when I'm encouraged to couple *my* computer with others without being offering the hardware equivalent of a condom. I think in the near future we may see users insisting on chastity, or at least serial monogamy, for their machines: *"Keep your mitts off my computer, bub! I don't share it with just anybody..."*

A Few Technology Assessments of My Own.

DOS: Like it or not, it's not going to go away. Like the committee members, I believe that 8086/DOS lacks some of the functionality required to support future applications; in particular, graphics are poor, the icky segmented address space is too small, and support for multi-tasking is inadequate. However, DesqView and Windows address the tasking issue, and the EMS memory kluge works around the memory restrictions; even though DOS users will tend to migrate to higher performance CPUs, they are unlikely to surrender hard-won familiarity with an OS environment unless they are offered something dramatically superior.

The Mac+ for a lot of us. Unlike the committee members, I believe that the level of functionality available in a Mac+ is adequate for the needs of most users; although it lacks some very desirable features, such as a large screen, memory protection, and a math coprocessor, it still has the requisite functionality, including network connectivity through LocalTalk and bus expansion using SCSI (which can be used to attach an Ethernet interface to connect to Cornell's higher performance Ethernet offering, for example).

As Adam Osborne said, "Adequacy is enough!" (he was mistaken only in assuming that a Z80 running CP/M was adequate, when its support for multitasking and graphics was terribly inadequate). If Apple had a user-service orientation rather than a bottom-line orientation, the Apple II series would be history, and the Mac+ would be *very* cheap (under $1000 *list*). Hopefully someday Apple will recover its collective memory and recall that the Mac was supposed to be a computing appliance—*and price it like one.* It may be a little on the slow side, but a slow computer is better than no computer. Well-written applications are reliable and deliver good performance.

As we look at systems more powerful than the Mac+, my preferences diverge even further from the recommendations of the Committee.

68030 Macs for colorful types. For users who need an easy-to-use machine with acceptable graphics performance *and* a low base price, or a machine with reasonably priced high-quality color systems, the Macintoshes are a very attractive choice. A color Mac costs about half as much as the average color workstation (though this will be changing rather rapidly). For users who do not really need color now, the choice between a Macintosh and other machines should be determined by your application needs and software availability. "High-end" Macs are *not* a particularly good deal when they are compared with comparably configured workstations.

OS/2? The doghouse is thataway. OS/2 has some interesting features, but a wheel is a wheel is a wheel, and the effort put into OS/2 would have been better invested in the domestication of UNIX, which, *mirabile dictu*, runs faster than OS/2 on Intel hardware (not to mention hardware OS/2 does *not* run on). Although some of the capabilities of the LAN Manager are noteworthy (particularly named pipes), most DOS users will find OS/2 expensive and overly complex. Because OS/2 is just another "traditional" operating system/network interface/graphics system/window manager with no special support for the needs of programmers, applications being developed for the OS/2 Presentation Manager will suffer from the same long-drawn-out debugging cycle as applications developed for the Macintosh.

UNIX: LIVE FREE OR DIE! OK, so I'm a little biased. I started learning C in 1980, and was introduced to UNIX in 1982. I then realized that the best thing about C is that it is *designed* to be the system programming language for UNIX, and that UNIX offers the same kind of laconic expressiveness as C. UNIX is unique because it was designed as a conjoined programming/operating environment that can run efficiently on a broad range of hardware; UNIX has survived because its conceptual basis is flexible enough to adapt to changing technologies; and there is good reason to believe that UNIX "will enter the year 1995 as the second-to-none, there-are-no-excuses choice" (Wayne Rosing, the Sun workstation and graphics technology VP, once the director of engineering for Apple's Lisa development project). UNIX (or perhaps I should say POSIX) is to operating systems as Fortran is to compilers: a universal if imperfect standard, which a vendor must either support directly or emulate.

Imaging systems: A unified display/printing environment is crucial to WYSIWYG. PostScript is a pretty good standard. Apple and MicroSoft are pursuing their own technology, in part because they believe it is superior, and in part because they can force it on their users, bringing them enormous profits. Aside from low-cost 32 bit color graphics, QuickDraw does not look so great these days; workstations offer much higher graphics performance than high-end Macs at about the same price.

Object-oriented development systems. IBM's recent commitment to support PostScript and NextStep could be very important. If IBM pursues a strategy of supporting *more* standards than any other vendor, it needs to guarantee a leadership role in some of the standard technologies; IBM definitely has the market power to make high-quality licensable technologies standards. NextStep in color on the RS/6000 platforms will be something to see. (Motif on X is nice to look at, but will not offer the same degree of WYSIWYG *printing* satisfaction, not to mention *programming* satisfaction.) The number and quality of the applications available for NextStep is impressive considering the brief period of time since its introduction, and that the software currently can run on *only one* machine, one which has a pretty high base price.

Modern computer systems are becoming too complex to manage with traditional lower-level programming techniques. The Macintosh is a prominent case in point: it is as much derogated for its programmer-fiendliness as it is praised for its user-friendliness. The complexity of its user interface, combined with an event-driven higher-level programming interface, lower-level interfaces requiring assembly language programming, and unforgiving hardware, has placed a disproportionately greater burden on the programmer, resulting in repeated failures to deliver products bug-free and on schedule. HyperCard demonstrates a partial solution to this difficulty, with a corresponding drawback: sluggish performance. MacApp provides another solution, albeit a proprietary non-portable extended-PASCAL solution.

MOTIF interfaces can provide a nice environment for users to work in; higher level tools like HP's New Wave are still needed to enhance programmer productivity, and some standardization needs to be brought about ensure progress.

My Own View of the Challenge.

Modern CPU/FPU and memory performance/price ratios will double every year, offering almost 1000 MIPS for a $15,000 workstation-class machine, or about 20 MIPS for a $2,000 "personal computer" with reasonably good color graphics and sound performance. Computer architectures will be liberally salted with high-performance chips optimized to perform specialized tasks.

Network P/P ratios will increase much more slowly, doubling perhaps every three or four years; unfortunately the cost of networking is determined in large part by the costs of pulling wire or fiber and making holes in walls, and there are no P/P revolutions in store for those activities. Despite rapid progress in the hardware field, demands for processing cycles, memory space, and network bandwidth will tend to outrun the hardware technology, due to:

- the voracious demands of very-high-level software systems;
- the rapid evolution and deployment of bandwidth-hungry mixed-media systems;
- user requirements that ease-of-use take precedence over performance; and
- market pressures to make popular applications available on a broad range of disparate platforms, leading to less efficient implementations.

To sum it up, we are going to be dissatisfied with much more powerful systems in the future; desires for feature-full systems will be whetted faster than they can be satiated.

The great challenge implicit in this forced-march evolution is to support the needs and desires of the well-funded advanced user without abandoning those who lack the money to purchase the latest and niftiest technologies. This requires a careful assessment of technological trends to predict the minimum levels of functionality required to support truly *productive* computing.

The draft report discusses the construction of first-rate digital highways and impressive high-end servers; users also need comfortable vehicles and worthwhile destinations. My reading of the report is that the authors want to restrict us to driving their favorite vehicles to their favorite places... even if we think the vehicles are hazardous and find that the service we get when we arrive is both overpriced and slow. We will recreate the suburban mishmash in a computing environment; just as in suburbia, you can drive all day for free, but you have to *pay* to be someplace interesting. This vision of the future contains no parks...

Cornell, along with other institutions in its class, has a unique responsibility deriving from its educational and scientific mission. We need to reconcile the computing needs of Nobel-oriented "grinders and visualizers" with the needs of other clienteles; not just Cornell graduate students struggling to make ends meet, or librarians coping with budgetary straitjackets, but also scientists, engineers, and business people working in the Third World, who will need to augment and draw upon databases, analyze technological and environmental trends, and communicate with others around the globe in order to address the urgent needs of clamoring billions within new constraints on the use of the "traditional" technologies developed by Western civilization. If these needs are not addressed, there will be no future at all for humanity, much less a future full of technological triumphs.

Teilhard de Chardin called the evolving global information-web the Nöosphere. The choices we make influence the development of this phenomenon; we can choose to focus solely on our most local needs and still improve our productivity and competitiveness, but I believe that only by examining a broader perspective can we ensure future prosperity for our institutions and our progeny.

Kevin Eric Saunders
cqu@cornellc