Manhattan College

Strategic Plan for Computing and Information Services
1998-2002

Addendum 1

Distributed Support for Computing and Information Services

Computer Governance Committee
September, 1998

DRAFT – FOR REVIEW ONLY

Manhattan College

Distributed Support for Computing and Information Services

"…Fundamental changes taking place in higher education make old models of information technology support inappropriate and insufficient. …" [1]

Introduction

"Two decades ago, our [computing and information services] users were a hardy group, knowledgeable about and seriously interested in computing. They were tolerant of system idiosyncrasies and failures. Adequate support meant posting signs in the computer center with examples of the control cards users needed in order to run different kinds of jobs. We wrote documentation and they had the motivation and expertise to decipher it. Recent users of information technology are often not particularly interested in the technology itself, and they are willing to spend only minimal time and effort to learn to use it. How we support the last 20 percent of the population that we are bringing into the technology environment is very different from the way we supported the early adopters. New users want "complete products."[2]

The situation at Manhattan College was and is no different than that described above by Moore. Twenty-five years ago, the engineering building housed behind locked doors and protective glass (now the EE dept.) the College's computer center. Keypunched cards were submitted by users or students who could (or were learning to) write their own programs. A central noisy impact printer placed their not-often-successful-on-the-first-attempt results onto wide greenbar paper and users returned later to pick up their rubberbanded output from a pigeonhole and try again. An early example of curricula support was freshman physics courses using "call labs" where student experimental data was keypunched and bundled with a header card that called up a pre-programmed lab #1 or lab #2 etc. as appropriate. Printed results, often in the form of charts and graphs made from dashes and asterisks, were then inserted in typed or handwritten student lab reports.

Administrative software was all home grown with most being Cobol programs authored by Br. William Hogan. Our "public" computing labs comprised one noisy keypunch room.

A move in the eighties to a renovated furniture store and warehouse - now the Research and Learning Center - relocated a somewhat more modernized but still central computer center

[1] Mclure, Polly A., John W. Smith and Toby D. Sitko.. *The Crisis in Information Technology Support: Has Our Current Model Reached Its Limit?* – CAUSE Professional Paper Series #16. 1997. EDUCAUSE. Boulder, CO.

[2] Moore,Geoffrey A.*Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers* (New York: HarperBusiness, 1991).

across Corlear Avenue. Control Data mainframes had been replaced by one of the then new generation of minicomputers – a DEC PDP 11/70. The old keypunch room had become a "terminal room" where users could stare at a blinking green or amber cursor on a video display terminal and call up programs to run.

Manhattan's first semblance of a campus network were an assortment of Micom boxes connected by dedicated data-grade phone lines into which a collection of "dumb terminals", mostly in administrative offices, were plugged. Some four years later a "packaged" administrative system (IA) had been acquired to supplement and then replace the home grown collection of Cobol programs (some didn't die easily, however and still function today) that automated administrative functions such as student registration.

By then the original IBM PC had been invented and ran under a text based operating system "developed" by William Gates, a young entrepreneur and MIT dropout. This new PC was heralded as breaking user chains to a central computer center. It had gone through several iterations leading to the XT and AT varieties which appeared in our early computing labs in DeLaSalle Hall (with tables customized by Br. Anthony Flynn) and the Research and Learning Center. These were labs comprising stand alone systems initially where printing was done via "sneakernet" i.e. you ran your 320 or 740 Kb 5¼" diskette (they were truly "floppy" then) over to a machine connected to the printer. And who among us here then can forget the infamous RLC rooms full of DEC Rainbow computers – supposedly IBM PC compatible but functional only when certain tides and sunspots were aligned (in retrospect perhaps an early warning system).

Visicalc which became the original Lotus 1-2-3 spreadsheet program was the first "killer application" (credited with launching IBM's PC) and for a time the only PC based software the College had to support in addition to compilers and programming languages. The IEEE had adopted an Ethernet 802.3 networking protocol and it was now possible to use thin and thick coaxial cable to connect computers together in a networked fashion. The DLS and RLC labs were so networked under an early version of a network operating system called Netware. Sneakernet gave way to print queues that worked most of time except when occasionally someone knocked off one of the coax connectors. Users downstream of the bad connection were all then disconnected from the network since the thin coax 10Base2 series wiring was not yet a modern star type topology.

Software grew plentiful and more sophisticated. Database applications followed the spreadsheets (Quattro had arrived as a competitor to 1-2-3 so of course higher education embraced both!) and soon DOS versions of Dbase, Foxpro and Paradox appeared on the College's fledgling network (why pick one when you can run all three!). Basic and Pascal and C – programming languages taught by the newly named Department of Mathematics and Computer Science – also needed to be supported. The social scientists had discovered computing as well so packages such as SPSS were added.

The comprehensive nature of Manhattan's curricula was to be mirrored in its supported software with specialized engineering, business and computer science applications competing for space on fileservers with basic office suites (three flavors at one time) and for limited support resources. While this evolution was taking place, a major mutation – the Internet and then World Wide

Web- appeared and demanded its own support. Academe, which spawned this creature, also fully embraced it and paradigms started shifting The IA management information system had become fully operational for student systems (recruiting, admissions, financial aid, registration), financial systems (bursar, controller), and college advancement systems (alumni, development).

Quietly, through the late eighties and early nineties, the Hayes Libraries gradually digitized its card catalog collection and eventually drawers full of typed and handwritten cards were discarded in favor of first an in-library terminal based searchable catalog and then a campus-wide network accessible online catalog. Connections to online search systems such as Dialog could be made initially only by a reference librarian. Searchable CD ROM collections were added and placed on a limited number of dedicated PCs for in-library searching. Links to online licensed databases were added after the Internet had spawned the World Wide Web which is today's link to the keeper of the College's online catalog and facilitator of campus wide network based research.

And so it was in the early nineties that an inextricable link between computing and information services and the academic curricula, library as well as administrative functions was forged. Computer technology had become entrenched in all prongs supporting the College's mission. Computing and information systems had evolved to become "mission critical" as in "if the network crashes and we can't get the bills out we're all up the creek" or "if I'm scheduled to teach MS Office in a computing lab, do I tap-dance for 55minutes if the server is down?"

As we evolved, most of the basics didn't change. A handful of networked computing labs grew to nine. The hardware evolved as the upstart computer industry sought to capitalize on the largest ever demonstration project of planned obsolescence by making the shelf life of new technology almost equivalent to that of yogurt. So 286s begot 386s which begot 486s which begot Pentiums which begot Pentium MMX which begot Pentium IIs. We connected more people to the network – most if not all faculty offices then the student residence halls. People moved and we reconnected them – and then they moved again necessitating further contributions to our wirer's retirement fund. Blue asynchronous copper wires begot coax that begot Cat3 twisted pair which begot Cat5 twisted pair which begot fiber. (Coax cabling disappeared here for all practical purposes shortly after a few well placed lightening bolts heated up the Capital Campaign and fried some electronics in Sears and Lavelle Halls. So some upgrades – such as the move to a fiber optic network backbone- were prompted by legitimate needs and sound planning as opposed to academe's usual "keeping up with the Jones's".)

When we exceeded two thousand network nodes, the beast was large enough to name and so the dawning of the age of ***Jaspernet*** arrived around the summer of '96. When the industry gave us faster networks higher education followed like lemmings to the sea and soon bridges were burned and replaced by routers and then switches and then Layer 3 switches and later gigabit capable switches. With these fast network infrastructures in place seemingly everywhere, E-mail emerged as a new communications vehicle – initially a novelty and now a necessity.

So, certainly not in unison and more like a cacophony than a symphony (where have you gone Lotus Symphony?), hardware, software, network equipment and wiring squeezed three or four generations worth of evolution into about a decade. Computer time surpassed dog years. And the

users came in droves – the early adopters, the late adopters, and the dragged-kicking-and-screaming adopters. They were faculty, administrators, staff and students and most of the three thousand or so of them needed support.

Mind you, most higher education curricula had not kept pace with technology development and there was no new crop of budding young specialists to recruit who were trained in modern software and networking. Industry figured this out quickly, set up certification programs and recruited our best and brightest CIS, EE and self-taught technology skilled graduates. They tempted staff with offers we couldn't match. Even sister academics raided our computing staff. Skills were learned mostly on the job and often by trial and error. Software that used to at least go through months of beta testing passed on that task to consumers. New versions every year or so gave way to monthly patches on vendor web sites and users wondered why the software didn't perform as promised or crashed in the middle of something. Surely, central-computing services was the culprit!

The Dilemma

So, the issue had been joined and the issue was support services or lack thereof. Who will keep the network from crashing, get us connected, set up out E-mail accounts, load and support the software we want to use in courses, facilitate online registration, arrange and support high speed Internet/Web access, teach us to use an operating system, help us format a letter, back up our data, fix our broken printers, recover lost files, kill all the viruses, show us a double mouse click and be faster than a speeding bullet and able to leap over software bugs in a single bound? Therein lies the dilemma. There were either limited or untrained bodies to turn to or nobody at all.

We've seen how we arrived at this predicament and that Manhattan is not unique in this regard. So, what do we do about it?

One model, proposed by McLure et al[1] in 1997 and based on earlier warnings by Yohe[3] and Gilbert[4], was presented to the Manhattan College Computer Governance Committee (CGC) at its Spring '97 meeting. At first the authors' notion of "distributed support" seemed plausible to the committee. Subsequent references to this concept at the Fall and Spring '98 meetings of the CGC drew more guarded reactions during discussion as it became clearer that "distributed support" meant that the computer center could not be all things to all users and that some departments – academic and administrative – may have to take ownership of certain computing and information services functions. So let's explore this concept as McClure *et al* sees it and then relate it and propose an adaptation of it to Manhattan College.

---

[3] Yohe,Michael "**Information Technology Support Services: Crisis or Opportunity?**" *CAUSE/EFFECT*, Fall 1996, 6-13.

[4] Gilbert, Steven W. *"Making the Most of a Slow Revolution: Recommendations from the AAHE Teaching, Learning, and Technology Roundtable Program," Change*, March/April 1996.

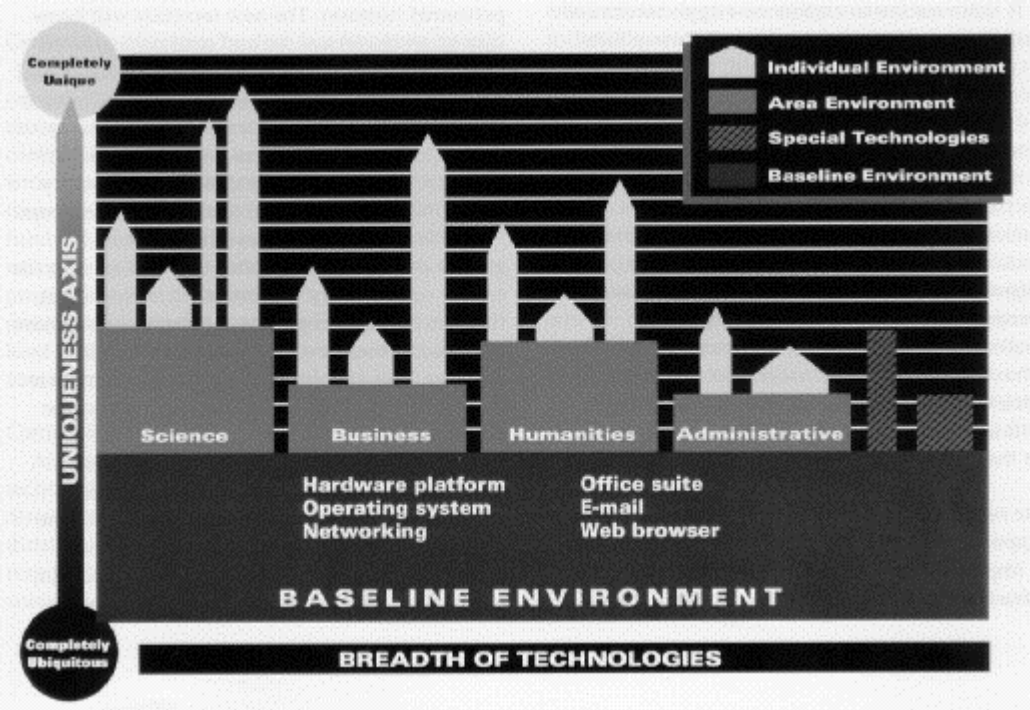
Distributed Support – The Model and Manhattan

Yohe[3] outlined well the expectations of users for which support was sought and the expectation is perfection.

> First, they expect that all information is immediately accessible to them; that they can have whatever information they want, whenever they want it, wherever they want it, and however they want it. We measure in minutes the time from experiencing delight that the library catalog is online to anger that the full text of all listed books is not instantly available on the screen.
> Next, they expect to have at their fingertip applications that will maximize their effectiveness. If it's available anywhere, it is expected to be available on their workstation. This implies that they will have the latest technology; that they will have connections to all other points and everything they reach will interoperate with their desktop hardware and software; that communication will be instantaneous, with integrated voice, video, and data; that the interface will be completely transparent; and that information will be independent of software or hardware. There is little understanding of the fact, for example, that a particular program will run only on one hardware platform, and even less patience with it.
> Third, they expect effective help in using these resources. We might postulate that this includes effective, conveniently available training and readily available, easily understood documentation. In fact, however, the real expectation is that the use of all of these resources will be intuitively obvious--with no recognition of the diversity of intuition that renders what is obvious to one person inscrutable to the next. They do not want to *learn* how to use these resources, only to *know* how.
> Fourth, if support is required, they expect to make a single contact, which will result in instant, on-site response by a person who has full knowledge of all hardware and software, from the very latest to the earliest legacy systems. Each customer's needs are naturally top priority to that person; the expectation is that they will be top priority to the support staff, too. Indeed, if a single customer has multiple needs, they are *all* top priority.
> Finally, our customers expect that there will be no problems: no network bottlenecks, no power outages, no downtime, not even any housekeeping or system maintenance time. Full service, twenty-four hours a day, every day of the year.

(anything with a familiar ring?)

McLure et al[1] refined Yohe's analysis by recognizing the following: **"The contemporary information environment is too complex and too interconnected for any individual or unit (departmental or central) to wholly conceive, manage, and maintain."** This in a nutshell is the crux of the problem. Caught up in the frenzy of a rapid pace of technology change and sometimes a desire to reap promised benefits overnight, unrealistic support expectations had arisen. Their "solution" - distributed support – can be fairly summarized by the hierarchical structure depicted in Figure A1.



1.1 Fig A1. Distributed Support Model of McLure et al

The lowest level represents the institutional information technology infrastructure. The functional capabilities in this level are made available to everyone in the institution. This environment is designed and managed to be highly reliable, consistent, and easy to use. Since this is an institutional environment, the central IT support organization is responsible for its management.

The technology support group can also sponsor special technology environments (the cross-hatched areas in the diagram). This is where the new technologies are introduced, explored, and tested. ...Thus we have a mechanism to migrate new technologies into the institution while not burdening all of the users with experimentation.

The second level represents areas or departmental standard environments. The primary definition and management of these levels are the departments, although there might also be coalitions formed for some areas that are multidisciplinary. Ideally, these environments build upon, rather than replace, the institutional standard environment. The extent to which a department customizes its environment is a function of its need for unique capabilities and its ability to design and support the additional features.

The third level represents individual user technology environments. Many people will be content with the institutional or departmental environment, hence will require no special support. Others will need or want an environment tuned to their specific needs. They will be expected to pay the cost of the uniqueness.[1]

The lowest level depicted above – the institutional information infrastructure – was defined as meeting "about 80 % of the needs of 80% of the users" and would include "**one** [emphasis added] of the office suites, a robust E-mail package and a web browser". Hardware would "run the software with reasonable performance, provide a consistent user interface, and be readily available."[1]

The analogy for Manhattan would be a fairly current version of the Microsoft Office Suite (presently Office 97); a campus-wide E-mail system that supports robust features such as

attachments, signatures, folders, lists, sorting, filtering, forwarding and POP/SMTP remote access (presently Pegasus mail v2.5); E-mail accounts available for all faculty, students, administrators and staff; and our browser choice - Internet Explorer (with Netscape Navigator still supported). Implicit in this baseline infrastructure is a high speed (at least T1) Internet connection and a 10/100 MBPS (gigabit capable) campus network with all of the necessary file servers, hubs, routers, switches, etc. that comprise our data wiring plant. Manhattan supplements McLure's academic orientated model by including in its baseline environment access to the IA system for deans' offices, academic advisors, designated faculty advisers, and department chairpersons. Administrative access would be read-only for most users with academic advisors and other authorized academic users having data input privileges as required. Manhattan also supplements their model by supporting a campus-wide site license virus checker/fixer software (presently Dr. Solomon's) as such a utility is deemed critical enough to be included in the baseline.

The above detailed package of baseline software, hardware and concomitant services would be the primary responsibility of the Computer Center to implement, maintain and update. It would constitute the Manhattan College institutional information technology infrastructure.

Manhattan does not presently have a functional equivalent to the area or departmental standard environment envisioned by McLure *et al* (their second level). It is a work in progress and an action item of this strategic plan addendum.

A review of the tables in the Strategic Plan for Computing and Information Services - 1998-2002 at pp. 46 –48 shows numerous software packages residing on central file servers which support individual departments or School(s). Examples are programming languages such as C++, Visual C++, Pascal and Visual Basic, older versions of the Quattro Pro and Lotus 1-2-3 spreadsheets and Excel, Paradox and Dbase databases still in use by some, and Wordperfect and WordPro – wordprocessors that are not part of the baseline Microsoft Office 97 suite. More specialized applications that, presently, are centrally supported include: I-DEAS, SPSS, Mathcad, Maple, Mathematica, Matlab, AutoCAD LT, Ed Scheme, ProII, Front Page, Fluent, MIA.

The distributed model envisions the user departments (or coalitions as appropriate where use is multidisciplinary) as having primary definition and management responsibility with management including financial responsibility. Presently, departments or Schools that lobby effectively enough for a particular software package to be supported do not assume any licensing, support or update costs for these packages. Specialized software and its support does not appear in departmental budgets and few departments acquire their own. Thus, there is no local financial disincentive to tapping a seemingly bottomless source of funding and no local financial tradeoff decisions need be made. (recall early CGC discussions of departmental "chargebacks" – an idea whose time may have come for Manhattan).

Software also doesn't function in a vacuum an often a part-time departmental specialist is needed to fully understand and instruct others in its use and explain its quirks (and there always are quirks). Updates and patches need to be monitored, acquired and installed, as appropriate and the technology transfer function updated.

Manhattan's essentially centralized version of this second level of the distributed model evolved because, in large part and with the exception of the School of Engineering with a .5FTE faculty member, not much local computing and information support existed at the departmental level. We have our share, of course, of the usual departmental gurus but software/hardware support, if done at all by them, was on a volunteer and irregular basis. It also evolved because some departmental computing needs had to be centrally supported *viz.* providing access for students in public computing labs to those "departmental" packages. The software marketplace also drove our development since in some cases it was more cost effective to have the computer center acquire a campus-wide site license from a vendor than have a department purchase single or multiple class section licenses (E.g. Borland).

Recall also that, as this evolution and information technology revolution was taking place, enrollments were declining and financial constraints precluded Manhattan and most of academe from adopting the departmentalized second level of the distributed model.

Manhattan's equivalent to the third level of the distributed model (essentially "every tub on its own bottom" for truly specialized computing needs) can be found in only a few departments. Environmental Engineering and Science has had a long standing need to support its computer modeling and simulation research – the mathematical water quality models pioneered here by Donald O'Connor. In the seventies, their research grants supported remote access to NYU's Courant Institute and provided capital support for remote card readers and printers. Today their grants provide support to acquire state-of-the-art personal computers and sophisticated laboratory instrumentation. The closest administrative analogy can be found in Institutional Advancement where the IA system has been supplemented by PC based fund-raising and donor tracking software such as Gifted Memory and Raiser's Edge acquired and budgeted locally. The support function has not been fully localized, however, further expanding the computer center's baseline.

Closing the Gap

Manhattan's challenge then is to close the gap between our present environment and the distributed model.

Yohe outlined the challenges posed as follows[3]:

- Providing sufficient connections, servers, routers, and hubs
- Providing sufficient capacity, in a multitude of dimensions.
- Coordinating efforts
- Integrating technology
- Keeping up with the pace of technology
- Securing funding
- Hiring and retaining qualified staff
- Attaining and maintaining peak efficiency and effectiveness

McLure *et al.* proposed the following strategy:

- Educate campus constituencies
- Engage users in decisions that affect them
- Redefine roles toward a federal model
- Create effective distributed support models
- Mentor individual staff (enlisting the enthusiastic support of some traditional technologists on our campuses)
- Recruit and replace, if necessary

Manhattan concurs with Yohe's recommendations and has embraced the first two prongs of McLure's approach through the efforts of the Computer Governance Committee, the Faculty Technology Committee and the IA Users Group. Creating effective distributed models (the federal model) is the work in progress alluded to above.

Action Agenda

It is proposed that Manhattan move as follows:

- Create and support an effective academic and administrative distributed model

    - Perhaps the School of Business with its quest for AACSB accreditation should be looked to as a first academic priority. Should faculty released time – akin to the School of Engineering – be provided to a "technologist" to be available as a local resource? Do the DLS based computing labs require dedicated support personnel? Can graduate CIS majors be tapped and funded as lab assistants and local on-call help desk personnel?

    - Student Financial Services may be the comparable administrative priority area and already has some "technologists" on hand.

    [Figure A2 is a draft update of the organizational structure that appears above as Figure 5 (p. 55, main body of Strategic Plan). It highlights some of the existing distributed support already in place in academic and administrative areas and proposes some additional positions in furtherance of this action item]

- Recruit the necessary staff to support the baseline institutional information technology infrastructure as defined above.
- Train staff to develop skills more appropriate to today's technologies
    - E.g. require CNE or MSE certification
- Expand and make more effective use of student support
- Continue software standardization initiatives to shrink the existing inflated baseline and require adequate justification for future acquisitions. Transfer the support

function for software that falls into the second and third levels of the distributed support model to the user departments – both academic and administrative.

- Transfer full or pro-rata costs of departmental or multidisciplinary software (again, at the second and third levels) to departmental budgets to promote standardization and ensure well reasoned cost-effective selections.

At the outset of this addendum, it was noted that old models of information technology are inappropriate and insufficient. Perhaps this agenda will provide Manhattan with the model necessary to better support its mission.