The Integration of Educational Computing in the
College of Humanities and Social Sciences

Preston K. Covey
Carnegie Mellon University, dtrollcovey@gmail.com

Joseph P. Devine
Carnegie Mellon University

Stephen E. Fienberg
fienberg@stat.cmu.edu

Christine Neuwirth
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/philosophy
Part of the Philosophy Commons
THE INTEGRATION OF EDUCATIONAL COMPUTING
In the College of Humanities & Social Sciences
Preston K. Covey, Director, Center for Design of Educational Computing
Joseph P. Devine, Associate Dean, H&SS
Stephen E. Fienberg, Dean, H&SS
Christine M. Neuwirth, Assistant Professor, English

CDEC Report # 88-17

July 1, 1988

A Version of the Report Prepared for the Lilly Endowment
Workshop on the Liberal Arts
Colorado College, Colorado Springs, June 18 - July 2, 1988

Center for Design of Educational Computing
Carnegie Mellon University
Pittsburgh, PA 15213
(412) 268-7641
The Integration of Educational Computing
In the College of Humanities & Social Sciences

OVERVIEW

1. PROJECT STATEMENT

   1.1 Background & Overview
   1.2 Technical Overview
      1.2.1 Hardware
      1.2.2 System Software
   1.3 Educational Applications
      1.3.1 The H&SS Core Curriculum
      1.3.2 Educational Applications in the Core

2. PROBLEMS

3. RESOURCES

   3.1 College & University Resources
   3.2 New Development Environments
      3.2.1 The CT Programming & Authoring Language
      3.2.2 HyperCard
      3.2.3 The Andrew Toolkit
   3.3 Existing Educational Applications

4. STRATEGIES & SOLUTIONS

   4.1 Faculty Training
   4.2 Development of Successful Models To Address Merit, Promotion, and Tenure Issues
   4.3 Student Training
   4.4 Communication Channels

5. EVALUATION PLAN
Overview

Carnegie Mellon University and its College of Humanities and Social Sciences have built, over the last decade, a tradition of leadership and innovation in educational computing. Now, we face another challenge and opportunity in this area that promises to have a major impact on the college's entire educational and research enterprise. Here we present a blueprint for the systematic and coordinated implementation of a major computer workstation and software development initiative intended to meet this challenge and opportunity. Experience has taught us that an initiative of this magnitude and import requires considerable thought and careful planning if it is to have a positive, integrated impact within the context of the college's technological, educational, and research environment. Our purpose is to provide information, anticipate problems and the resources needed to address them, and to outline the college's strategy for solving these problems and achieving the goals of this initiative.

We begin, in Part 1, with background on the college's experiences with educational computing development and applications initiatives over the last decade, and follow with a technical overview of the hardware (in the form of the Macintosh II) and software that will come to dominate the college's educational computing environment. Part 1 ends with a description of the college's core curriculum and educational computing applications (past and anticipated) in the core. In Part 2, we discuss problems (both technical and non-technical) that accompany a major initiative of this sort, while, in Part 3, we focus on college and university resources and new development environments available to address these problems. Part 4 delineates the college's strategies for integrating this computing initiative with its educational enterprise, addressing issues of faculty and student training; development, testing, implementation, evaluation and dissemination of successful models; and attendant communication channels within the college necessary to maintain this initiative. Finally, in Part 5 we describe the college's plan for evaluating the impact and assessing the success of this initiative.

We hope that this paper will serve as the touchstone to which we look for reference and direction as we venture into this area with high aspirations for the college in all aspects of its work.
1. PROJECT STATEMENT

1.1 BACKGROUND & INTRODUCTION

Carnegie Mellon University's College of Humanities and Social Sciences (H&SS) has an exceptionally rigorous and integrated core curriculum, the foundation of university core requirements, designed to provide its students with the intellectual tools required to understand today's world. These include tools of literacy and numeracy, of historical analysis and social analysis, of cognitive science and computer science, as well as concepts, perspectives, and knowledge in a wide range of fields. The college itself is divided into six departments: English, History, Philosophy, Psychology, Social and Decision Sciences, and Statistics, with Economics being offered by the Graduate School of Industrial Administration and Modern Languages under the auspices of History. All of the college's units play an active role in the core curriculum.

In 1982, H&SS was the first college at Carnegie Mellon to provide personal computers to virtually all of its faculty for use in both teaching and research. As a result H&SS faculty began to develop class-related applications in numerous disciplinary areas. The use of computers in the core curriculum, however, has been limited, in part because most faculty in the college have had access to only the lowest level of personal computers. A piecemeal approach to acquisitions and application and combined with limited access to high-level, advanced function workstations has attenuated the educational impact of personal computing on the curriculum. To enhance further the pedagogical impact of computing in the college's core curriculum, we require a more integrative approach which inevitably will be embedded in an increasingly more powerful computing environment.

At the same time as the college began to support the use of personal computing in its educational programs, the university initiated the development of a campus network structure based on higher-level computer workstations named Andrew, after our founder, Andrew Carnegie, and subsequent benefactor, Andrew Mellon. Andrew consists of three levels of service: network services and communication (e.g., mail, bulletin boards, printing, the library information system), file service, and Unix-based user interface software. Andrew's development and its day-to-day operation comes under the responsibility of the Academic Services division (the central university unit that provides computing services to the campus). The development of Andrew is sufficiently advanced to allow the university to proceed towards a goal of workstations on the desks of all faculty members and students, in a flexible and varied fashion.

H&SS is taking the lead in introducing workstation computing in undergraduate education. The combined research resources and development tools available in the new workstation environment will make it far easier for faculty to provide students with good interfaces, perspicuous access and tools for 'hands on' exploration of real data and research agendas and, thereby, to integrate research into the educational agenda.

The college plans to introduce workstation computing across the undergraduate curriculum using Macintosh II computers. During this summer, as a result of a recent agreement with Apple Computer Inc., 60 Mac IIs will be provided to H&SS faculty for the purposes of educational development, and the university will install an additional 25 Mac IIs this fall for student use in public clusters in the college's facilities and in the adjacent library. Beginning in Fall 1989, H&SS will recommend that all incoming freshmen students buy their own computers.
(specifically Macintosh workstations), much in the same way as textbooks are recommended for courses. All of the college's computer workstations (both faculty and student) will be connected to Andrew. The intent is that, once powerful personal computers are commonplace across the college, they will achieve an impact on education which will greatly exceed the effect of piecemeal acquisitions and applications.

H&SS plans to develop and use a variety of new and enhanced computer-based educational tools for its core curriculum. These will enable students to explore the computer as an instrument for learning independent of formal course structures. Essential to this effort is the implementation of an educationally-sound support system for the use of computer-based tools and the Andrew network by both students and faculty. This will include the targeting of resources, the planning of training programs for faculty and students, the generation of software, and an evaluation program to examine the impact of this new computing environment on our curriculum and the educational process more broadly. While this effort begins with the core curriculum, the college expects the impact to extend to upper level courses and to further integrate the education-research mission.

This project is a collaborative one between H&SS and Academic Services, and will impact not only H&SS and its faculty and students, but also the other academic divisions of the university for whom Academic Services provides computing resources. A successful development project may also serve as a model for other institutions.

1.2 TECHNICAL OVERVIEW

1.2.1 HARDWARE

The principal computer in H&SS for education will be the Macintosh II. Apple's grant consists of 60 Mac II workstations, each configured with 4 MB of memory, an 80 MB hard disk, a 19 in. monochrome or 13 in. color display, an Apple standard keyboard, EtherTalk, and a mouse. In addition, Apple will provide 4 extra 80 MB disks configured with A/UX, the Apple Unix operating system (4 of the grant machines may also be configured with A/UX), and 4 Compact Disc Read-Only Memory (CD-ROM) units. CD-ROM's hold relatively vast amounts of information (software, data, text, graphics, audio, video). For example, one resource, available now in the college on CD, is the complete Oxford English Dictionary. Other information is becoming available on CD as well, such as encyclopedias and census data.

The 60 grant machines are being deployed to departments for faculty use with the following allocations:

English - 13
History - 10
Modern Languages - 3
Philosophy - 6
Psychology - 6
Social and Decision Sciences - 8
Statistics - 6
College Staff - 6

The special hard disks with A/UX and the CD-Rom units will be installed in a central H&SS computing resource laboratory which will house 6 or 7 Mac IIs (2 from the Apple grant and 5
from Academic Services), an optical scanner (a machine for scanning printed material into the computer), and a variety of other manufacturer's workstations. During 1988/89 printing facilities will be provided over a college local area network linked to at least 8 laser printers.

A number of faculty around the college already have Mac IIs, Microvaxes, Sun's, or other powerful workstations acquired in connection with research activities. Specifically, Philosophy, Psychology, and Statistics have their own local area networks of workstations and computers. These college networks will all be linked in some form prior to the planned linkage of the MAC IIs to the Andrew network in the summer of 1989.

1.2.2 SYSTEM SOFTWARE

The Mac IIs will eventually be able to run two independent sets of system software. The first is the standard Macintosh software. The second is the Andrew software, developed at Carnegie Mellon in a related project, which is based on Unix. H&SS will be the site of the first deployment and test of the MAC II Andrew Software.

During 1988/89, H&SS faculty and students will be able to use Mac IIs to run applications under the Macintosh operating system or under A/UX alone. The Andrew software, based on A/UX, will be available on Macintosh II computers in late 1989. A beta-test version will be available sooner and interested faculty will test it and provide feedback to Academic Services.

1.3 EDUCATIONAL APPLICATIONS

The central part of the project will be a coordinated drive to integrate educational applications in courses across the college, especially in the core curriculum. These applications will run under either the Mac OS or Andrew, whichever proves to be most convenient.

1.3.1 THE H&SS CORE CURRICULUM

The H&SS core draws on all of the college's departments as well as others (e.g., computer science, the physical and natural sciences, and the fine arts) outside the college. Students enter the core curriculum as freshmen and exit after their sophomore year as they enter a major.

The curriculum is arranged in five thematically-organized clusters:

1. **Fundamental Problem Solving Methods and Skills.** Five-six courses (one in expository writing, one in computer programming, one in the theory and application of cognitive psychology, one in philosophic methods of thought and reasoning, and one or two in statistical methods and analysis). The dominant theme in this cluster is the variety of paradigms across disciplines that are central to understanding and practicing problem-solving. These courses are mostly concentrated in the freshman year, so as to facilitate application of these skills in subsequent courses. In addition, all students enroll in a Computing Skills Workshop, designed to introduce them to the computing resources and environment of the university, including the Andrew network.

2. **Humanistic and Social Values.** Four courses, (one in philosophy - the same course
listed above, two in history, and one in literary and cultural studies). The philosophy course applies methods of philosophic reasoning to issues of social values and ethics. One history course is designed as a comparative civilizations course, while the other (chosen from an approve list of five) emphasizes the methods and central concepts of social historical analysis in the study of American history. The literary and cultural studies course introduces students to both the literary and cultural contexts of "texts" of various sorts: selected "great works," contemporary best sellers, and such "cultural texts" as advertising and television linking three domains: the study of language, the study of history, and the study of culture.

3. **Social, Political and Economic Systems.** Three courses (two in the social sciences, and one in microeconomics). The first social science course focuses on the analysis of decision making as a behavioral and social phenomenon while the second places these principles of decision theory into the context of American political institutions. They are followed by a course in microeconomics, which views economics as a social and behavioral science linked to complex and often irrational patterns of decision making.

4. **Language, Culture and the Arts.** Three courses, chosen from a distribution of options that span three areas of study: languages (with options in the study of a modern language, or the study of linguistics); cultural studies (with options in western culture, non-western culture, technology and culture, and religion and culture); and the arts (the theory and practice of one or more art forms, including courses offered by the College of Fine Arts).

5. **Science and Technology.** Two courses in a physical or natural science, or one course in science and a second in a related technology that makes serious application of scientific principles. In addition, about 2/3 of the college's students take a two semester calculus sequence in their freshman year.

Opportunities abound for integration of knowledge and methodologies across disciplines in the core curriculum. Increasing use of shared technology further enhances these possibilities. The clustered arrangement of courses in the core keeps instructors and students from operating in an intellectual vacuum or in relative isolation. Courses are already connected to larger conceptual structures in ways that make each cluster more than simply the sum of its parts, and that encourages multi- and inter-disciplinary collaboration and perspectives.

1.3.2 **EDUCATIONAL APPLICATIONS IN THE CORE**

Educational applications of workstation technology include the following activities:

- Evaluation and use of existing tools (e.g., statistical packages, word processors, spreadsheets, databases, electronic mail, bulletin boards).

- Development of computer-based course materials (e.g., online syllabi, lecture notes, readings, data).

- Use of existing educational tools and packages (e.g., simulations of societies or laboratory experiments; intelligent tutoring systems).

- Development of easy-to-use user interfaces and tutorials for existing research tools (e.g., a user interface for state-of-the-art statistical research software; a multi-media document that
Development of new tools and packages.

The college's objective is to have a mix of these educational application activities in each course in the core curriculum. The following, specific activities are additional targets:

Development in each of the six departments of one showpiece application which can be distributed outside the university. (Some of these applications are already under development in some form, e.g., The Great American History Machine.)

Development of at least one application which makes integral use of network communications (e.g., English Writing Tools).

Development of at least one application which uses interactive video disk (e.g., The Right to Die).

A careful evaluation of available statistical packages for the Mac OS.

Faculty developing new educational applications as part of this project will be strongly encouraged to distribute these applications to other universities through the Academic Courseware Exchange. Because the goal of this project is to influence the direction and use of computing both at Carnegie Mellon and elsewhere, the documentation, evaluation and widespread distribution of all innovative applications developed are important objectives.

2. PROBLEMS

Despite 20 years of hope and hype, the computer revolution for education is disappointingly late in arrival. This section identifies problems that have tended to inhibit the development of innovative applications of computing in education. The following sections identify institutional resources and strategies for addressing these problems.

Problem 1: Lack of faculty knowledge. Faculty members doing research in their disciplines typically command the knowledge needed to carry out the enterprise. In contrast, faculty who wish to produce educational applications require knowledge from a number of interdisciplinary sources. In order to produce an application program with the potential for wide distribution, both within the college and beyond, faculty may need support for one or more of the following activities:

- obtain outside grant funds.
- carry out pedagogical and design research.
- detail the functional requirements and create design specifications for software.
- program.
- write documentation.
- create graphic designs.
- obtain software licenses and copyright permissions.
- maintain software and hardware on which the application is based.
- design and carry out a formative evaluation of the software (e.g. user & classroom testing).
- supervise the project.
- design and carry out a summative evaluation of the effectiveness of the software.

Few faculty members possess the breadth of knowledge required to carry out all these activities effectively, nor should they have to acquire it.

**Problem 2:** Development costs to faculty in time, effort, or money are too high. A major difficulty and disincentive posed for faculty innovators is the cost associated with development. For example, a book or several papers could be produced in the time it takes to develop a well-designed, substantive educational application with demonstrable impact. Even applications that do not begin from ground zero, such as adding an interface to an already existing research tool, can be time-consuming. Most effective educational applications require large amounts of well-designed graphics and interaction with the student. Until recently, both have been notoriously difficult and labor-intensive to do well, even if the faculty member is an experienced programmer - which the vast majority are not, do not want to be, and should not be obliged to be.

**Problem 3:** Limited dissemination potential. A faculty member who produces a book takes the potential for widespread dissemination for granted. In contrast, educational applications are often extremely limited in their dissemination potential. Until recently, conventional programming tools have been tied to particular machines. Software developed in such conventional programming languages on one kind of hardware will not necessarily be compatible with other types. In particular, graphics and interaction for educational applications must, as a rule, largely be reimplemented to make a program run on other machine types. As a consequence, while computers are increasingly used in higher education both for teaching computer use or skills and for routine academic work with commercial software (e.g., word processing), they are rarely used for direct instruction. Some high quality educational applications exist, but are not disseminated widely because of their dependence on a particular type of computer and operating system.

**Problem 4:** Uncertain merit; uncertain promotion and tenure standards. Compared to more traditional research and educational activities, it is less certain what criteria and standards of achievement should be applied to evaluating educational applications and their contribution to research and education. Departmental, college and university promotion and tenure committees have no clear or generally agreed upon guidelines for evaluating educational software developments by their faculty. Even if a particular unit develops guidelines, untenured faculty have good reason to be wary of an effort that will likely consume significant time but carry little or unpredictable weight for career advancement. The same may be said for tenured faculty, although the possible negative consequences for the advancement their careers are less dire.

Despite these problems, several faculty in H&SS have already developed exemplary application software running on other machines or in different computing environments. Examples include (1) *The Great American History Machine*, developed in a college core course and the recipient of an EDUCOM/NCRITICAL Distinguished Software award and (2) the *LISP Tutor*, an intelligent tutoring system for teaching the LISP programming language that dramatically improves students' achievement over traditional classroom instruction. The Mac IIs provide an opportunity both for increased student access to these exemplary application programs (when they are converted to run on the Mac II) and for new initiatives by faculty with ideas for innovative educational applications.

The gains that can be achieved in the quality of student education makes the strategic
development of high quality educational applications a major college priority. Serious efforts to encourage such development need to be linked with solutions to these problems. In section 3 we identify the resources that are available for overcoming the problems and in section 4 we identify strategies for addressing them.

3. RESOURCES

3.1 COLLEGE AND UNIVERSITY RESOURCES

In the previous section, we identified the wide range of activities required to develop successful educational applications and noted that most faculty lack the knowledge to carry out all of them. This section identifies college and university resources for supporting faculty in carrying out these activities.

H&SS and Academic Services will support the equipment to be used by faculty, including the 60 grant machines from Apple. This support will be provided in a variety of ways and will include software and communications interfaces for grant machines, maintenance of grant machines, printers and related supplies, software licenses for discipline-specific software, as well as staff support and guidance.

The current college-level computing staff includes a Director of Computer Services and two support staff that maintain the college's current resources. Before Fall 1988, the college plans to increase its team of people who support computing by the addition of a new senior staff member to oversee all of the college's computing efforts and three programmers for the educational computing project. This staff support is in addition to the existing research staff in the college's departments. The new programmers will support the development of educational computing projects in the departments and supplement existing departmental research staff.

The college will provide along with each Macintosh II computer a basic set of software which runs under the Mac OS. The exact list of software is still being determined and depends upon the ability of Academic Services and H&SS to negotiate suitable site licenses. The standard software will include something resembling the following basic list:

Macintosh system and multi-finder
Hypercard
word processor (possibly with low, medium, and high level options)
graphics editor
CT (formerly CMU Tutor)
MacGnome
Macintosh network package
statistics package (Systat and/or Minitab)
relational database
spreadsheet
programming languages (Fortran, Common LISP, Pascal)

In addition, there is a long list of applications which we hope to provide, preferably under both the Macintosh and Andrew operating environments.

Academic Services will provide support for other parts of the project including placement of
machines and software in the public clusters. To stimulate and coordinate these developments, the Center for the Design of Educational Computing (CDEC), a part of the Academic Services division, will assign support staff to the project.

CDEC has three functions relevant to the project: (1) research and development of model educational applications, (2) support for faculty developers, and (3) educational services. As a research and development center, CDEC brings together faculty and research scientists from across the arts and sciences who have academic expertise, technical expertise, and experience in the design, development, and deployment of educational applications as well as research interests in instructional and interface design, cognitive science, or learning theory.

CDEC selects areas of strategic need and projects with seminal impact. For example, CDEC produced the CT programming and authoring language, a cornerstone in our strategy for the support of widespread applications development. Seminal applications propagate uses across several disciplines and provide models for the integration of educational computing in both research and curricula. For example, PD World, a simulation environment built in CT for developing Prisoner-Dilemma models of the evolution of cooperative behavior, allows the replication or extension of all extant research (thus putting a serious research tool in the hands of students as well as faculty) and is used in courses in philosophy, political science, decision science, economics, social psychology, and sociobiology. CDEC thus brings interdisciplinary resources, state-of-the-art knowledge and disciplined focus to bear on the complex and otherwise disparate issues in designing and integrating educational applications of advanced technology.

CDEC provides support for faculty developers in several forms, which vary with the case and available resources: direct consultation on design and development, grant proposal development, and course deployment; professional and student programmer support; provision of new development tools (several besides CT); as well as research conferences, technical seminars and training workshops. CDEC staff work with H&SS faculty as well as University Studies on the development of curricula that integrate computer applications in general education.

On the educational service side, CDEC is currently responsible for the Computing Skills Workshop course, and the CDEC Scholars Program in Educational Computing, which supports students who work with faculty and other university units on the development of computer software and documentation. CDEC Scholars also serve as training tutors for faculty (which has included Carnegie Mellon's President).

3.2 NEW DEVELOPMENT ENVIRONMENTS

CT, Hypercard, and the Andrew Toolkit are development environments which we expect faculty to utilize in the development of new courseware and disciplinary based application programs. All three languages have been designed to solve many of the problems that lead to high costs in time, effort, and money and that limit the dissemination potential of educational applications.

New development tools like CT, HyperCard, and the Andrew Toolkit make it expedient for faculty to build user-friendly interfaces, interactive instructional modules, graphic data analysis and plotting programs, and other educational tools on top of powerful state-of-the-art research programs. The memory and processing capacities of the new workstations support advanced research as well as educational applications; this means that faculty can use the same
systems as their students, for research. Data storage and retrieval is further enhanced by access to distributed files on the Andrew network.

3.2.1 THE CT PROGRAMMING AND AUTHORING LANGUAGE

CT is well adapted to creating instructional interfaces for research tools and classroom aids. Specifically, CT embodies several desirable features for a modern development environment with educational developers uppermost in mind.

CT is a programming language that is easily learned and readily usable by non-experts. Features making CT easy to learn and use include:

(a) **On-line documentation** allows scrolling, cross-indexed, complete documentation to be searched and read in its own window on the screen. Sample code illustrating CT functions can be clipped from the help window, pasted to the source code window and then run in the execution window. The user can alter this sample code to experiment with different programming functions and their effects.

(b) **A dynamic graphics editor** allows the user easily to draw desired displays directly on the screen in the execution window. The corresponding code is produced automatically in the source code window. Direct manipulation of graphics or objects in the execution window automatically rewrites the code appropriately in the source window.

(c) **Incremental compiling** allows the developer to immediately see the results of new code, but the program always runs at compiled speed. Both are crucial for non-expert programmers who make many mistakes and need to see results quickly, for quickening the learning curve as well as the iterative design-implementation-evaluation cycle.

(d) **Automatic error checking** is facilitated by a menu with which the user can select common commands for automatic error-free input, and an interactive syntax checker that locates and diagnoses incorrect input immediately in the source code window.

CT provides easy control of modern computers' interactive capabilities and special interface features, including answer-judging, resizable windows, multiple and 'fancy' fonts, pop-up or pull-down menus, interpretation of text and mouse input, and intermingling of text and graphics.

For the most popular modern personal computers, CT is hardware and operating system independent. It currently runs on the Macintosh and Mac II, the IBM AT and PS/2 series, and various Unix workstations. A program can be written or run on any of these machines, with no alteration or translation processes whatsoever. Furthermore, each machine runs its own native operating system (Mac OS, DOS, Unix, etc.) so that use of CT does not interfere with other work.

In order to exploit and build upon programs implemented in other languages (e.g., to build an instructional interface, on-line documentation, etc. for an expert system written in LISP or a research program in "C"), CT has hooks to other languages (e.g., LISP) and packages (e.g., HyperCard). This allows one to do in either language what is best done in that language (e.g., inference engines in LISP, dynamic graphics interfaces in CT). The languages with which CT communicates or interacts will be increased to include Fortran and others, thereby allowing the porting and upgrade of extant research and instructional software to more modern computers and environments.
CT is a programming language with many modern features and more planned for the future. It supports object-oriented programming and production rules. Therefore, although CT can easily be learned by a beginner, it does not limit the work of experienced developers. Sketch, an "intelligent" tutor for graphing algebraic equations and the largest program written in CT, won a 1987 EDUCOM/NCRIPTAL Distinguished Educational Software Award. CT has also proved useful to researchers, who use it to plot results or to build time-saving interfaces to old or badly designed research programs (e.g., for access to super computers).

CT is largely complete. A beta-test version for the Macintosh and Mac II is currently available, with a full version expected by Fall 1988. Production quality CT for other workstations will follow during the 1988-89 academic year. While enhancements will continue, CT already provides a machine-independent, non-experts' educational programming environment powerful enough for experts as well.

3.2.2 HYPERCARD

Like CT, HyperCard has been designed to make the production of high quality computer applications easier. HyperCard allows faculty and students to create multi-media, interconnected, interactive documents. Multi-media documents include not only text and static, graphic elements, but also computer animations and computer-generated audio. With additional hardware, computer-controlled video and audio are also possible. Multi-media documents can help students visualize abstract concepts and employ active learning strategies. For example, multi-media documents can be constructed that allow statistics students to rotate a three-dimensional model of multivariate data while reading related material (this would be a variant on the commercial package, MacSpin).

Interconnected documents allow faculty and students to create cross-references among related items, references that "link" the items together. Items that are linked together can be accessed quickly - with the click of a mouse button - without searching the library stacks and without losing the original context. Interconnected documents can help students see connections between a text and its historical, political, philosophical, religious, aesthetic or social contexts. For example, faculty and students can create links from phrases and illustrations in a document to other literature, illustrations, annotations, and responses. With access permission, teachers and students can view each others' responses and create links in the same document. Interconnection can enhance traditional teaching tools as well. For example, faculty or students can link elements of a syllabus to detailed lecture notes for each session.

Interactive documents allow readers to move text, graphics, and other elements; add, delete and modify elements; and change the state of elements. Interactive documents can be constructed to filter information for particular readers and to help readers locate information. For example, an interactive syllabus document could be constructed that would add class assignments from each syllabus for students' courses to students' online "To Do" lists, providing students who use it with a time management aid.

In addition to a graphics-based, user-friendly authoring interface, HyperCard provides an easy-to-learn, object-oriented programming language. It is also extensible, with an interface to C and Pascal. HyperCard runs only on upper-end Macintoshes and Mac IIs.
3.2.3 THE ANDREW TOOLKIT

The Andrew Toolkit has two primary goals: (1) to support the development of stand-alone applications that integrate text, graphics and images in a standard, efficient user interface; and (2) to support the development of multi-media editors, that is, editors that allow users to edit text, equations, graphs, tables, pictures, and so forth, all in a single program.

Like other advanced function workstation documents, an Andrew Toolkit document can contain both a document text and a picture. Unlike documents based on most user interface toolkits, however, an Andrew Toolkit document integrates the text and picture while retaining the desirable functionalities of each. Although text and picture in the same document appear to be a single object, the text and the picture are actually separate objects that can be manipulated independently. For example, users can pop-up menus for editing text in the text region and menus for editing the picture in the picture region.

In addition to allowing users to edit different types of objects in one place, the Andrew Toolkit supports the development of application programs that can include arbitrary objects upon demand. For example, ez, the Andrew Toolkit multi-media object editor, can dynamically load any object that has been created according to the Andrew Toolkit protocols. The editor does not need to know about the object in advance. This capability supports the integrated development of additional tools (e.g., music editors, dynamic outliners) that go beyond standard ones (e.g., word processors, spreadsheets).

The Andrew Toolkit has been designed to maximize the portability of applications. It currently runs on numerous advanced function workstations (IBM RTs, SUN 2s & 3s, MicroVAXes) under two window managers (X.11 and the Andrew window manager, wm) and will be ported to MAC IIs. The Andrew Toolkit has already strongly influenced the developing, industry-wide standard for user interface toolkits.

3.3 EXISTING EDUCATIONAL APPLICATIONS

H&SS and Carnegie Mellon are not alone in the development of quality educational software. If the workstation initiative is to be successful, we will need to take full advantage of materials and applications developed elsewhere, integrating them with home-grown materials. Statistical software packages represent a good illustration of the issues here. They are often elaborate, require enormous resources to develop, and they need to be examined from the alternative perspectives of what statistical procedures they incorporate, the user interface, the numerical analysis underpinnings of the computations. Few universities, including Carnegie Mellon, will want to invest in the development of such packages, but the evaluation of statistical packages for use in the Carnegie Mellon environment, both in teaching and research, is a crucial task that requires high levels of expertise. Statisticians familiar with existing packages and their strengths and weaknesses need to be willing to assist the college with the evaluation and the selection of externally-available software.

In virtually every area of the college, one or more faculty will be encouraged to focus on the examination of existing educational applications and to begin to develop network contacts to make Carnegie Mellon a possible test site for materials currently in preparation.

Yet another dimension to the issue of existing educational applications, is the adaptation of
these applications to course-related settings in the college. The emergence of CD-ROM as an important medium for data and image storage presents a good illustration. For example, while resources as diverse as the artwork in the National Gallery and the Oxford English Dictionary are already available on compact disc, someone needs to develop the interface to make them accessible for classroom use. Similarly, the results of the 1990 Decennial Census down to the zip-code level will be released on compact disc (in dBase II readable format) roughly five years from now, and this data set has the potential to be an unparalleled resource for a variety of statistics and social science courses. We need good data extraction, manipulation, and display interfaces to make effective use of this resource.

Creative uses of the data resources and interfaces developed by others represent an important component of the transformation of the core curriculum and thus form an important component of this project.

4. STRATEGIES AND SOLUTIONS

With an initiative that directly involves over fifty faculty at the outset, we cannot expect everyone to launch into the development of elaborate and new educational applications. Rather we want to approach this project expecting a mix of computer-related activities with some faculty doing high-level development, others preparing small course-specific modules, and yet others thinking about the uses of already available material. Our goal is to change the educational environment and thus we need all of these components if we are to be successful. The strategies and solutions that follow should facilitate this goal.

4.1 FACULTY TRAINING

Training in the use of the Mac IIs and the basic applications software will begin in July 1988 for the recipients of the H&SS grant machines. The college's Director of Computing Services will present a series of workshops over the summer on the use of existing applications programs.

Developer workshops for the use of CT and HyperCard are planned for Carnegie Mellon faculty and students and special workshops are planned for H&SS College faculty during the Summer and Fall of 1988. These will begin with half-day introductions to these development environments, followed by more detailed introductions to their use.

As with any course material, producing HyperCard stacks or CT materials requires an investment of time and in some cases money. Multi-media, interconnected, interactive documents require time to prepare. Most faculty and students have little experience in generating high-quality graphics. For some documents, the assistance of professional graphic artists may be desirable. The greatest amount of time may not be in the preparation of such documents, however, but rather in the time spent thinking about ways in which these capabilities can enhance the goals we have for teaching and learning, and the evaluation of their effectiveness. College resources will be targeted at common support materials for developing applications in these environments. For example, staff are currently developing a HyperCard "shell" that can be used to prepare an online course syllabus for each course in the core in a generally common format.

Faculty training will continue in an informal manner with activities and resources for experimentation collected together in the college's new computing resource laboratory.
4.2 DEVELOPMENT OF SUCCESSFUL MODELS TO ADDRESS MERIT, PROMOTION, AND TENURE ISSUES

The uncertainty regarding merit, promotion, and tenure standards for educational computing applications can be addressed by a strategy to develop successful models of such applications. Such models should help faculty planning to undertake the development of new computer materials. The following is a starting point for elaborating those models, a process that will require additional involvement from faculty.

One model is to embody ongoing research ideas in the educational application that are publishable separately. For example, the *LISP Tutor* is a test-bed for the implications of the ACT* theory of skill acquisition. Even though the development of the *LISP Tutor* has taken approximately ten person years, numerous technical reports, conference presentations, and research papers, published in proceedings, edited volumes, and journals, have appeared.

This model, based to some extent on the "proof of concept research" in computer science and other engineering disciplines, is not without difficulty in some arenas. Many disciplines, especially the arts and humanities, value developments in educational computing only to the extent that material is available for their students to use and not as mainstream research - if they consider it research at all. Much computer-based educational development, however, lends itself to interdisciplinary collaborations that can combine educational and research goals. For example, faculty in Music collaborated with faculty in Computer Science in order to produce a musician's workbench for composing. Similarly, faculty in English have collaborated with colleagues in Social and Decision Sciences on the evaluation of the *Comments* program, a tool designed to facilitate response to writing. Such collaborative work not only tends to increase the quality of the research product and process by which it is made, but it often increases the number of audiences for the project.

A second model involves an analogy to a textbook. Degree of merit is awarded according to the rigors and prestige of the publication's review and venue: the better the journal or press, the better the schools that have adopted it, the better the publication. As the portability across operating systems and hardware enabled by such programming environments as CT and the Andrew Toolkit increases the dissemination potential of educational applications, this model is likely to see increasingly widespread application.

The textbook model also has potential difficulties. Publication is neither a necessary nor a sufficient condition of merit. Many of the most intellectually sophisticated applications make use of the most advanced technology, for which there is no installed base beyond a few privileged campuses; much of the published software for which there is a significant installed base of current or low-end technology is correspondingly pedestrian in intellectual content and function.

A third model involves the direct inspection and evaluation of faculty-developed software by peer review committees. For example, several programs (*The Great American History Machine*, *Sketch*) have been reviewed by external committees and received awards.

Peer review is complicated by the existence of at least two relevant 'peer' communities: (1) other developers who are experienced in the technology and the attendant design and development issues (who may hale from various disciplines and therefore not be optimally critical - or sensitive to issues in one's own discipline); and (2) other members of one's own discipline (who may not be sophisticated in the technology, sufficiently critical about pedagogical or educational...
issues, or sufficiently imaginative about the possibilities).

The bottom line is that for some applications, the intellectual task of developing an application and the demonstration of its value to the university community may have to be regarded in their own right, especially in connection with promotion, tenure, and salary deliberations. Standards have been developed to some extent for creative production (e.g., plays, works of art, fiction) and technical production (e.g., construction of high-energy accelerator components and other equipment needed to carry out research). The needed standards for evaluating computing applications, as well as overall respect for the hybrid enterprise, will come with the increased knowledge and further education.

4.3 STUDENT TRAINING

Students will clearly require training on two related but subtly distinctive levels: technical and attitudinal. The provision of such training will be a part of the college initiative, and not something left to individual courses to handle. Extensive student computer training within the structure of the course for which the applications have been designed is highly disruptive, diverting time and attention from the substantive content of the course for the purpose of detailed, fundamental technical training.

Technical training needs are relatively clear and straightforward. We need to teach the students how to use the Macintosh II, some common applications software, as well as the Andrew computer network environment. The principal vehicle for this training will be the existing Computing Skills Workshop (CSW), that all incoming freshman and transfer students take during their first semester. The aim of the CSW is to introduce new students to the full range of computer hardware and software common to the university computing environment to which they will have access, place these applications into a general educational context, and then train students to minimal levels of competency in the use of these machines and applications. This course will continue to serve the key training function of providing a systematic and well-rounded introduction to the university's diverse and sophisticated computing environment, and leaving them minimally functional with the technology.

The challenge of "attitudinal training" is more subtle but no less crucial. The key here is socializing students to see the technology and attendant tasks as fundamentally supportive of primary learning agendas, but at the same time both complementary and integral (vs. peripheral, intrusive gadgetry). Here again, development and implementation of educational computing applications must be driven by the intellectual mission of the curriculum and not simply by the availability of powerful computer hardware and software. Careful attention must be paid to how students meet and interact with these applications in (and across) their course work in the core curriculum. It is essential that all faculty and teaching assistants involved in any way in delivering courses with such applications be thoroughly trained in both the purpose and methods of these technology applications and communicate those purposes to students.

4.4 COMMUNICATION CHANNELS

The college's centralized curriculum governance model promotes the development of communication channels within and across numerous levels as well as providing an important type of quality control. Central sponsorship and coordination promotes symmetric integration as
contrasted with inchoate invasion of the curriculum by applications that are quixotic, diversionary, and gimmicky. This model allows for the encouragement of entrepreneurship, but with structural filters in place to screen applications before they are incorporated into core courses. This filtering allows the college to act as a clearinghouse of information and ideas about educational applications, pointing core course faculty and departments toward potentially useful applications or collaboration with applications in other disciplines.

The college's Academic Advisory Center occupies a strategically opportunistic position to monitor and study student usage of computing, external to individual courses, and in ways that examine educational impact both microscopically (within individual courses) and macroscopically (across courses and the curriculum). In addition, the Center provides a central link with faculty whose expertise is relevant. This could include, among others, faculty in the social and behavioral sciences who study the social and organizational impact of computing, and faculty in writing and cognitive psychology who study the impact on learning processes of different pedagogical tools and techniques. Finally, the Center is strategically well-placed to participate in the coordinated collection of longitudinal data regarding the impact of computing technology on student learning.

Communication channels are important at student levels as well. Technology available on the Carnegie Mellon campus facilitates information exchanges in some ways that are new and experimental, and in others where we have had substantial experience. As an example of the latter, the university's computer network capabilities make possible computer bulletin boards through which knowledge is shared among students (and between faculty and students) about what applications are available, about advertisements offering or seeking tutoring assistance, and the like. The establishment of special bulletin boards for core curriculum activities will be an early priority of the workstation project.

5. EVALUATION PLAN

The project will be evaluated jointly by H&SS and by Academic Services, and overseen by a joint steering committee. The committee consists of H&SS faculty and staff from both the college and Academic Services and is assisted by an advanced graduate student from the Department of Statistics. The evaluation process is beginning with the careful articulation of measurable objectives for the project. The actual evaluation will include the distribution of a series of questionnaires to students and faculty, as well as direct and indirect measures of the impact of educational computing activities and resources. Baseline statistics will be collected during 1988/89 and statistics on impact will be collected the following year.

A set of preliminary faculty questionnaires are currently being distributed to the faculty, prior to the installation of the grant machines. Even those faculty not receiving grant machines have been asked to complete a subset of the items on the questionnaire so that appropriate comparisons can be made between the developments by the faculty directly related to the first phase of the Mac II project and the development of materials by other faculty.

The joint steering committee will prepare quarterly progress reports during 1988/89 that will detail both software development and evaluation materials.

Careful evaluation is difficult when we attempt to disseminate an innovation across a diverse population in a short period of time. The evaluation of the workstation initiative needs to provide
convincing evidence of the value of a coordinated support infrastructure and special incentives of the educational computing enterprise and, more importantly, the value of educational computing to the education of our students and faculty. As we embark on this project, input from faculty that helps us to refine our goals and mechanisms for assessing whether we meet those goals is of crucial importance.