9-5-1988

Advanced Computer Applications for The New Liberal Arts

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

Follow this and additional works at: http://repository.cmu.edu/philosophy
Part of the Philosophy Commons

This Technical Report is brought to you for free and open access by the Dietrich College of Humanities and Social Sciences at Research Showcase @ CMU. It has been accepted for inclusion in Department of Philosophy by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.
Advanced Computer Applications for
The New Liberal Arts

Report on Grant # 83-10-1 to Carnegie Mellon University

Preston K. Covey
Vice Provost for University Studies
September 5, 1988

A Final Cumulative Report
Covering the Four Year Period from July 1, 1984 to June 30, 1988

Submitted to the

Alfred P. Sloan Foundation
Suite 2550
630 Fifth Avenue
New York, NY 10111

On Activities Supported by a Grant to Carnegie Mellon University
For the Development of Educational Applications of Computing
in the New Liberal Arts

Grant # 83-10-1

Amount of Grant: $500,000

Grant Period: October 7, 1983 to June 30, 1988
### Summary Contents

**Executive Summary** 5

1. The Goal of the Sloan Foundation Grant 5
2. The Scope of the Final Report 6
3. The Organization of the Final Report 6

**I. Overview & Highlights** 7

0. The Center for Design of Educational Computing 7
1. Research & Development 8
2. Technology Transfer & Dissemination 15
3. Education & Integration in Curriculum 17

**II. Strategic Initiatives & Projects** 19

0. The Center for Design of Educational Computing 19
1. Productivity Tools 23
2. Model Software 33
3. Multi-Media Environments 50

**III. Financial Report** 65

1. Budget Summary 65
2. Budget Detail 66

**APPENDIX: CDEC Technical Reports 1984-88** 68
## Detail Contents

**Executive Summary**

1. The Goal of the Sloan Foundation Grant 5
2. The Scope of the Final Report 6
3. The Organization of the Final Report 6

### I. Overview & Highlights

0. The Center for Design of Educational Computing 7
1. Research & Development 8
   1.0 Awards to CDEC Projects & Personnel 8
   1.1 Productivity Tools 9
   1.2 Model Software 10
   1.3 Multi-Media Environments 11
   1.4 Research Agendas 12
2. Technology Transfer & Dissemination 15
3. Education & Integration in Curriculum 17

### II. Strategic Initiatives & Projects

0. The Center for Design of Educational Computing 19
   0.1 Why the Focus on Design? 19
   0.2 Interdisciplinary Infrastructure 20
   0.3 Why an R&D Center? 20
   0.4 CDEC's Mission & Measure 20
   0.5 Criteria for R&D Projects 21
1. Productivity Tools: Solving Problems for Developers 23
   1.1 Common Lisp & AI Tools 23
   1.2 Multi-Media Tools 23
   1.3 The cT Programming Environment 25
      1.3.1 Development Problems Addressed 25
      1.3.2 How cT Solves these Problems 27
      1.3.3 Availability & Use of cT 29
      1.3.4 Historical Note 31
      1.3.5 References 31
Advanced Computer Applications for the Liberal Arts

2. Model Software: Trail Blazing From CAI to AI & ICAI

2.0 The Carnegie Mellon Andrew Project: Background for Model Applications 34

2.1 The Fine Arts:
   2.1.1 Architecture 34
   2.1.2 Art 35
   2.1.3 Music 36

2.2 The Natural Sciences & Mathematics:
   2.2.1 Biology 36
   2.2.2 Chemistry 37
   2.2.3 Physics 38
   2.2.4 Mathematics 40

2.3 The Humanities & Social Sciences:
   2.3.1 English 41
   2.3.2 History 45
   2.3.3 Philosophy 45
   2.3.4 Psychology 48
   2.3.5 Social & Decision Sciences 49

3. Multi-Media Environments: New Media for Values Education

3.1 Some Observations & Provocations 51
   3.1.1 Lest Questions of Evaluation Evade Judgments about Values 51
   3.1.2 Neglected Domains: Open-Ended Inquiry in the Arts & Humanities 52
   3.1.3 Why is Surgery Taught in Theaters, but Ethics Only in Theory? 53

3.2 What is Multi-Media? 55

3.3 Why Multi-Media? 56

3.4 CDEC Strategy & Priorities for Multi-Media 57

3.5 Project THEORIA: New Media for Values Education
   3.5.1 A Right to Die? The Case of Dax Cowart 61
   3.5.2 Art or Forgery? The Case of Han Van Meegeren 62
   3.5.3 Birth or Abortion? The Human Face of a Dilemma 63
   3.5.4 Values Boggled: Ethics, Art & Money in the Work of J.S.G. Boggs 64

III. Financial Report

1. Budget Summary 65

2. Budget Detail 66

APPENDIX: CDEC Technical Reports 1984-88 68
0. Executive Summary

0.1 The Goal of the Sloan Foundation Grant

The ultimate goal of the grant from the Sloan Foundation to Carnegie Mellon University was to promote the quality and use of advanced computing in liberal arts education; an instrumental goal was to establish a resource center for doing so. The establishment of a research and development center was central to the project: to build a 'critical mass' of technical and academic expertise to provide a strategic vision for enhancing the 'new' liberal arts through advanced technology.

In this context, the New Liberal Arts is meant simply to emphasize the full spectrum of modern arts and sciences: quantitative, technological and computer literacy as well as the qualitative, humanistic understandings traditionally associated with a liberal arts education. This broader understanding of the liberal arts has been emphasized at Carnegie Mellon from the beginning.

The projects supported under the Sloan Foundation grant encompass the sciences, mathematics and the fine arts as well as the humanities and social sciences -- with a keen eye to the need for computer literacy and innovative computer applications in the more traditional liberal arts as well as the more technical and quantitative disciplines. Our projects address the needs of general science education as well as formal and quantitative work in traditional humanistic fields such as philosophy and history. We have also applied computer technology in striking and perhaps surprising ways to humanistic fields such as ethics and aesthetics -- where neither formal theory nor quantitative apparatus typically apply, but where rigorous methods of evidentiary analysis and hypothesis testing must prevail nonetheless.

Overall, our projects respect a certain irreducible reliance of all liberal arts studies on methods of analysis, synthesis and hypothesis testing that can be facilitated by appropriately designed computer tools. We have attempted to demonstrate the benefits and applicability of computer technology to a full complement of liberal arts disciplines.

In addition, we inaugurated a university-wide course in computer literacy for students, which has evolved along with our computer environments, from time-sharing to personal to distributed computing with a diversity of microcomputers and advanced workstations. We devised a program of computing-related curricula. And we have addressed strategic problems for faculty developers of educational software in an age of programming-intensive and incompatible computing systems.

The Sloan Foundation has not merely supported a proliferation of (40-odd) model projects but has also promoted a strategic vision for enhancing the liberal arts through advanced computer technology. Two principles have guided our selection and development of educational applications:

1. An application should be strategic: address pandemic educational problems or crucial learning bottlenecks, promote intellectual skills not well promoted by traditional means, effect improvements in the learning process with the computer that cannot be accomplished without the computer, and force us carefully, critically and creatively to rethink, reaffirm or revise our basic concepts, pedagogy and goals.

2. An application should be seminal: propagate use, impact and benefits in more than one domain or discipline, serve as a paradigm for propagating needed innovation in educational strategy or curriculum, and exploit or advance important disciplinary or educational research.
0.2 The Scope of the Final Report

The Overview & Highlights and Strategic Initiatives & Projects sections of this report describe the outcomes of the Sloan grant and CDEC activities for the full period of the grant. The Financial Report section is organized by Carnegie Mellon fiscal years and covers the four-fiscal-year period from July '84 through June '88. Fiscal year 1984-85 (July '84 - June '85) overlaps with the first periods of Sloan funding reported by my predecessor as CDEC Director, Jill Larkin (September '83 - October '84 and November '84 - December '85).

The Sloan Foundation grant was originally to end June 30, 1987. It was extended to June 30, 1989. We have acquitted the Sloan funding effective June 30, 1988. This report, therefore, constitutes a final report on the grant, with financial detail covering all but the initial period of September '83 - June '84 reported by Jill Larkin in her first report of December 12, 1984.


0.3 The Organization of the Final Report

Part I provides a brief Overview & Highlights. Part II more fully describes the Strategic Initiatives & Projects of the Sloan grant, organized by the major strategic categories of the Center for Design of Educational Computing's activities. Our most strategic projects are described at length and provided with references to technical reports or publications that describe them in detail. While all the Sloan-supported development projects are catalogued in Part II, this part also elaborates upon our three major R&D initiatives: productivity tools, intelligent tutoring systems and simulations, and multi-media environments.

Part III is a Financial Report for 1984/85 to 1987/88 with a Budget Narrative organized by the categories in the Budget Summary and the Budget Detail spreadsheets at the end. This section explains the rationale for the allocation of funds for the 40-odd Sloan-supported projects.

There is an Appendix listing the Technical Reports produced by the Center for Design of Educational Computing since fiscal year 1984-85. This is one measure of CDEC's growth in productivity since its first full fiscal year of Sloan funding. Any of these reports (or videotapes) can be obtained from the Director for further information on Sloan-supported or other projects.

I am also submitting with this report a copy of a strategic campus report that I co-authored on The Integration of Educational Computing in the College of Humanities & Social Sciences and a shrink-wrapped copy of The cT Programming Language package for the Macintosh, which contains the cT book and user manual as well as the software.

The cT programming language (formerly CMU Tutor) is perhaps the most strategic and seminal product of CDEC and Sloan funding, since it addresses key problems of productivity among faculty developers. The strategic importance of cT / CMU Tutor for the development of educational software is corroborated in "New Strategies for the Development of Educational Software" by Mark Resmer, then Director of Academic Computing at Vassar (Academic Computing, December 1987/January 1988 -- copy also enclosed).
I. Overview & Highlights

The Sloan grant supported about 40 projects (including documentation and curriculum as well as software development) and over 50 software programs (excluding system versions), in addition to establishing the Center for Design of Education Computing itself.

In the course of the Sloan grant activity Carnegie Mellon's computing environment evolved from mainframe time-sharing to personal microcomputers to distributed computing involving multiple (over 100) types of mini-, micro- and advanced-function computers on a sophisticated campus-wide network. CDEC has tracked and kept pace with this evolution in its design, development, deployment and dissemination efforts -- and reached beyond it to develop multi-media environments for both microcomputer and advanced-function workstations.

CDEC software development has been integrated with core curricula and university-wide educational initiatives as well as regional and national dissemination efforts.

0. The Center for Design of Educational Computing

The single most important achievement to be credited to the Sloan Foundation grant is the founding and flourishing of the Center for Design of Educational Computing (CDEC) itself.

CDEC's mission is to provide leadership and support for the design and development of advanced computer applications for the liberal arts. CDEC aims to generate strategic planning, design wisdom, dedicated academic and technical expertise focused on the problems and promise of advancing technology for the liberal arts.

CDEC has distributed both substantial funding and equipment grants for faculty across campus with strategic and seminal impact in mind. CDEC has spawned and supported 75% of the educational computing applications developed on campus and 95% of the advanced workstation applications of the last four years. (See CDEC Productivity Report 1984-88, CDEC Report # 87-17).

Technology transfer, dissemination, faculty support and direct educational services have all been important to CDEC's mission. But the crucial factors in our success have been our focus and selectivity, the way we have built expertise and concentrated resources over time in certain strategic areas and seminal projects: there is a long-term strategy at work here that would not emerge from a less dedicated, less focused, less interdisciplinary, less venturesome enterprise.

CDEC's strategic activities and impact encompass three categories:

1. Research & Development
2. Technology Transfer & Dissemination
3. Education & Integration in the Curriculum

Because the discretionary Sloan grant has made both strategic and adaptive planning possible and because CDEC's strategic vision is itself an important outcome, I have organized the first parts of this report according to these categories of strategic initiative. The following are some highlights of the three categories of CDEC activity. The strategies and projects themselves are described more fully in Part II of this report..
1. Research & Development

1.0 Awards to CDEC Projects & Personnel

Only in the last two years has there been a national awards program for software in higher education, a joint effort between EDUCOM (Princeton, NJ) and NCRIPTEL (The National Center for Research to Improve Postsecondary Teaching and Learning at the University of Michigan, Ann Arbor). All the CDEC software submitted to this or any competition have won awards, including one of CDEC's interactive videodiscs:

David Miller and Stephan Greene's *The Great American History Machine (GAHM)* received a "Distinguished Software" award in the 1987 EDUCOM / NCRIPTEL competition.

Jill Larkin, Carol Schectic and David Trowbridge's *Sketch*, an intelligent tutor for graphing algebraic equations built in cT (CDEC's own programming environment), also received a "Distinguished Software" award in 1987.

Ruth Chabay's *General Chemistry* software won the "Best Chemistry Software" award in 1987. (Ruth was recruited to CDEC from Stanford University in 1986-87.)

David Trowbridge's *Graphs & Tracks*, a simulation environment and intelligent tutor for introductory kinematics, won in both the "Best Physics Software" and "Best Integrated Software" categories in 1988.

Preston Covey, Scott Roberts and Steven Bend's prototype videodisc *A Right to Die? The Case of Dax Cowart* received a "Certificate of Merit" in the 1988 University of Nebraska Interactive Videodisc Awards competition.

Both CDEC videodiscs (developed only in the last year), *A Right to Die?* and *Art or Forgery?*, will be entered in the 1989 EDUCOM / NCRIPTEL competition in March, as will *PD World*, the *CMU Proof Tutor* and *Symbolization Tutor*, below.

Jill Larkin, CDEC's founding director, received a Guggenheim Fellowship during 1986-87 to advance her work at the intersection of cognitive science and instructional design. Preston Covey, CDEC's director since 1986, chairs the American Philosophical Association's committee on computing. All CDEC research staff are leaders in their respective academic fields in educational computing.

Besides the award-winning software, many of CDEC's 50 programs -- more than the ten originally proposed in the Sloan grant proposal -- are leading, state-of-the-art applications. All of the following examples (as well as the other Sloan-supported software) are used in courses at Carnegie Mellon and many are used at other institutions as well.
1.1 Productivity Tools

CT (formerly CMU Tutor), in the category of general programming and authoring environments.

CDEC's premier 'shrink-wrapped' product for distribution is the CT programming language and environment for the Macintosh. (A copy of the Macintosh version is included with this report. IBM's Academic Information Systems is distributing CMU Tutor, the prototype of CT developed for Unix workstations. CDEC itself is distributing all versions of CT.)

To address a basic productivity problem, how to get faculty eventually to generate a broad, rich applications base, we have addressed the heart of the problem: the need for better, easier, more sophisticated productivity tools for implementing innovations in increasingly sophisticated environments. The CT programming language is the prime analogue (although it is not the only example). CT is easy to learn, easy to use -- relative to any extant general programming language; it makes it easy to control the modern interface features of advanced computers (which make modern programming all the harder for novices and experts alike); and it is machine and operating-system independent, allowing its applications to run unaltered on the most popular microcomputers as well as advanced Unix workstations, thus obviating the hardware incompatibility problems that are a major disincentive to faculty developers.

CT was designed to address certain strategic problems for faculty and student developers of educational applications, namely:

1. Difficulties or inefficiencies in learning and use.
2. Lack of support for modern graphical interfaces.
3. Lack of portability or compatibility across hardware and operating systems.

One singular achievement of CT is portability: the language and its applications run without change on popular (IBM and Macintosh) microcomputers as well as a variety of Unix workstations (Sun, DEC Micro VAX and IBM RT). Another is compatibility over time: CT / CMU Tutor programs written three years ago at its inception still run, despite changes in hardware and operating systems, thanks to automatic source code conversions.

Unlike conventional authoring languages, CT is a flexible general-purpose programming language; it trades off high-level authoring for programming capabilities and extensibility, yet its "single most important virtue . . . is undoubtedly its accessibility to new users" (Mark Resmer, then Director of Academic Computing, Vassar College).

The strategic significance and impact of CT is discussed in Mark Resmer's "New Strategies for the Development of Educational Software" (Academic Computing, December 1987 / January 1988).

CT is described more fully in Part II, section 1.3, below.
1.2 Model Software

Besides good productivity tools, the field of educational computing needs good model applications that set standards and exemplify what it means to be strategic and seminal in the use of advanced computer technology in the liberal arts.

Our applications attempt to track and advance the evolution of strategically important types of application: sophisticated simulation, graphics and data-analysis environments and 'intelligent' computer-assisted instruction (ICAI) and tutoring systems (ITS) that exploit research in artificial intelligence and cognitive science. Among the showcase Sloan projects are:

**PRINCE**, in the category of 3-D graphics.

**GAHM**, in the category of graphical data-mapping and analysis.

The *Notes* and *Comments* programs, in the categories of critical writing, reasoning and collaborative learning environments, along with upgrades of *Writer’s Workbench*.

**PD World**, built in cT, in the category of simulation environments and research tools for both students and faculty, for building, studying and testing n-person iterated Prisoner’s-Dilemma-type models of the evolution of cooperative behavior and collective rationality.

**Cognitive Psychology Labs** (14 programs), in the category of simulation labs. These programs are used in the psychology core curriculum course and published by Brooks Cole. They have evolved from mainframes to microcomputers and are currently being reimplemented on the advance-function workstation in cT.

The *Lisp Tutor*, an expert system for teaching structured programming in Lisp. (See John R. Anderson and Brian J. Reiser, *The Lisp Tutor*, Byte, April 1985.)

**VOICE**, a music-theory and composition tutoring program with synthesizer interface.

**DEMOS**, an expert decision modeling system for assessing subjective probabilities and risk, and **Giraffe**, a graphing environment for DEMOS et al.

**TETRAD** et al., a suite of expert-system programs for teaching probability analysis, statistical and causal modeling, and inductive logic. (The suite of AI and inductive logic software is integral to the Logic & Computation Program, which offers a major and a minor as well as graduate degrees in the Department of Philosophy, an interdisciplinary program that bridges formal logic, the philosophy of science, mathematics, statistics, social, cognitive and computer science. TETRAD itself is a state-of-the-art expert system for generating and testing competing causal models on very large data bases in the social sciences. See Clark Glymour, Kevin Kelly, Richard Scheines and Peter Spirtes, *Discovering Causal Structure: Artificial Intelligence, Philosophy of Science, & Statistical Modeling*, Academic Press, 1988.)
Graphs & Tracks, a simulation environment and intelligent tutor built in cT for introductory kinematics. Winner of the “Best Physics Software” and “Best Integrated Software” awards in the 1988 EDUCOM / NCRIPTAL Higher Education Software Program.

Sketch, an intelligent tutor built in cT for graphing algebraic equations. Winner of a “Distinguished Software” award in the 1987 EDUCOM / NCRIPTAL Higher Education Software Program.

ANALYTICS, a package of three CAI programs for teaching first-order logic and its applications to argument analysis and logical problem solving in natural language. This package, used in the philosophy core curriculum course, has evolved from mainframes to microcomputers and the advanced-function workstation environment.

CSYM, an intelligent tutor for translating between the artificial language of first-order logic and English.

The CMU Proof Tutor, an intelligent tutor for the construction of natural deduction proofs in first-order logic. The Proof Tutor incorporates an expert system for proof construction designed on human heuristics that is also provably ‘complete’ for propositional logic. It is built in Common Lisp with a graphical interface built in cT for incorporation into an on-line computer-managed course in first-order logic.

These 30 programs reflect the types and range of applications we have supported with Sloan funding. The projects are described along with the 20 other Sloan-supported software projects catalogued below under Model Software in Part II, 2. Programs for interactive videodisc are described separately under Multi-Media Environments in Part II, section 3 of this report.

1.3 Multi-Media Environments

Multi-Media here refers to any interactive computer-based environment that dynamically links several media: text, graphics, animation, sound, full-color high-quality video (motion or still). Multi-media environments are singled out for special emphasis and attention, even though they also represent ‘model applications.’

No strategic vision of the evolution of educational technology can afford to ignore the promise and problems posed by rapidly evolving multi-media technology (currently, interactive videodisc and CD-ROM technology, although these particular laser-optic technologies are not definitive of the rapidly evolving state-of-the-art).

No strategic vision of educational computing can afford to ignore potential applications of this technology in the largely neglected domains of the arts and humanities, those ‘soft’ areas of open-ended value-laden inquiry that seem to resist practical and rigorous methodologies delivered even by traditional means or media. Curricula and pedagogy in these domains have never been so evidently important or so arguably in need of improvement.

For these reasons, CDEC has made a strategic priority of tracking and addressing the promise and problems of multi-media technology and has chosen the problematic staging area of values
education for doing so. In the fullness of time, we will expand our efforts into the areas of science and technology education that are already exploiting computer-based multi-media.

CDEC's current focus in multi-media is a series of projects under the rubric Project THEORIA, whose agenda is reflected in its acronym: Testing Hypotheses in Ethics/Esthetics: Exploring the roles of Observation, Rationality, Imagination, & Affect. The goal of Project THEORIA is to design compelling, interactive simulation environments for testing hypotheses and 'theories' of the arts and morals -- among the most difficult and disputed of human value domains.

Theoria (Greek for theory) is also an allusion, to the concept of theory rooted in concrete observation, to the etymological roots of both theory and theater in the ancient Greek verb theorein: to see, to view, to behold. Through exploitation of multi-media technology, we aim to provide a theater for ethical and esthetic theory, to bring the theory to ground in realistic settings that are rich in the complex data that any competent theory must first behold in order to explain.

For the theory and teaching practice that informs Project THEORIA, see:

Covey, Preston K. Project THEORIA: Interactive Video Media for Ethical Reasoning. CDEC Report # 86-01.

Four projects are presently in various stages of concept, design or development for ultimate delivery on advanced workstations and microcomputers:

A Right to Die? The Case of Dax Cowart (a videodisc, first in a series, which won a "Certificate of Merit" in the 1988 University of Nebraska Interactive Videodisc Award competition)

Art or Forgery? The Case of Han Van Meegeren (a videodisc, which has and will be demonstrated at several national education, art, museum and philosophy conferences)

Birth or Abortion? The Human Face of a Dilemma (mixed media, with videodiscs, presently in prototype development)


CDEC's multi-media agenda and projects are described more fully in Part II.

1.4 Research Agendas

Research of various sorts is part-and-parcel of our development activities. Research is needed in the design of productivity (programming and authoring) tools, in the design and management of modern interfaces and interactive mechanisms; on development strategies for educational computing products and programs; on the pedagogical, cognitive and learning theories that inform interface design or expert tutoring by any means; on the expert knowledge required to build intelligent tutoring systems in our chosen domains; on the underlying technologies themselves; on the subject matters, methodologies and skills we aim to enhance with these technologies; and on the institutional and conceptual issues that attend educational or curricular reform that implicates computer technology. CDEC's selection and promotion of model applications projects is not ad hoc
or merely opportunistic, but seeks to promote a broader understanding of the intellectual and educational issues that they entail.

Special emphasis is placed on instructional design research and modern interface design. There are generic design and implementation lessons to be learned from our work that should help propogate research, design and development strategies beyond the shelf-life of any particular application. These lessons and our research agendas are reflected in the books, papers and technical reports by CDEC PI's or research staff listed below, all of which either motivated or were generated out of Sloan-supported projects:


Chabay, Ruth, Jill Larkin and Bruce Sherwood. A Computer Based Curriculum to Teach Qualitative Concepts in Electricity. For the AAAS Symposium on Qualitative Reasoning in Physics, January 1989.


Covey, Preston K. Multi-Media Technology & Values Education. Presidential invited address for the 30th International ADCIS Conference, November 1988.


2. Technology Transfer & Dissemination

Technology transfer includes the dissemination of expertise, understanding and research on educational applications of computing as well as the dissemination of hardware, software and hands-on training. CDEC aims to provide leadership on campus, regionally and nationally in these functions through various organizations for which it is headquarters or in which its staff participate as leaders. While CDEC's priority has naturally been higher education, our concern with general education in the liberal arts broadly construed has embraced projects targeted at both secondary and adult education.

Sloan funding has made it possible to build and support the kind of academic and technical infrastructure required to provide the outreach and technology transfer activities below.

Besides the scores of demonstrations and presentations that CDEC Pi's and research staff give at other schools and national conferences each year, CDEC's major efforts in technology transfer and dissemination are focused through the following mechanisms and organizations:

CDEC is headquarters for the InterUniversity Consortium for Educational Computing (ICEC). ICEC members include representative colleges and universities across the country. ICEC publishes a newsletter, maintains a national electronic network.
CDEC provides ICEC with office space, administrative support, conference and workshop support, and consulting.

CDEC originally founded and continues to provide technical, logistical and consulting support to the Consortium for Computing in Undergraduate Education (C-CUE), a consortium of small liberal arts colleges in Pennsylvania and West Virginia.

CDEC is headquarters for the Pittsburgh Interactive Learning Forum (ILF), a regional consortium of schools, corporations and agencies concerned with educational applications of laser optic technology.

CDEC is a charter member of the Research Consortium of the National Demonstration Laboratory for Interactive Video Technologies (NDL) at the Smithsonian Institution. Through the NDL, CDEC staff consult on educational projects using laser optical technology.

CDEC organizes workshops for and collaborates with star teachers from regional high schools through the Allegheny Intermediate Unit (AIU). CDEC has provided cT workshops for high school teachers developing educational software and the Sketch program was the result of collaborative development and classroom testing with the AIU.

CDEC is de facto headquarters for the American Philosophical Association's Committee on Computer Use in Philosophy, by virtue of Director Preston Covey being chair of the committee. CDEC co-sponsors the annual national Philosophy & Computers Conference, now in its third year, and will host the conference in 1989. CDEC co-sponsors the educational software fairs held in conjunction with annual APA meetings.

CDEC edits and publishes Computers & Philosophy quarterly, an interdisciplinary journal with an interdisciplinary audience and contributors from the fields of philosophy, mathematics, computer science, psychology. C&P publishes articles on educational computing, artificial intelligence research, cognitive science, and curriculum as well as software reviews and general academic computing information.

CDEC is headquarters for the McDonnell Foundation Program in Cognitive Studies for Educational Practice, which administers a research grant and post-doctoral fellows program.

CDEC sponsors national conferences and conference sessions, such as (most recently): the 1988 McDonnell Foundation invitational conference on Artificial Intelligence & Education, whose proceedings Computer-Assisted Instruction & Intelligent Tutoring Systems: Shared Issues & Complementary Approaches are edited by CDEC staff and will be published by Lawrence Erlbaum and the LaserActive '88 conference session and volume of proceedings Laser Optical Media in Higher Education which are organized and edited by CDEC staff.

CDEC projects and staff have a very high profile at national conferences. For example, at the August 1988 meetings of the Conference for Philosophy & Computers and the Conference on Computers in Physics Instruction, CDEC research staff gave 10% and 20% of the presentations, respectively. CDEC's national visibility is apparent from our Technical Report series. On campus, CDEC has served technology transfer in the following prominent capacities:
CDEC funded and established the university's Educational Software Library in 1986.

CDEC has administered four major university-wide hardware deployments (microcomputers and advanced-function workstations) for faculty developers (over 200 machines) since 1985.

CDEC provides technical and consulting support to Carnegie Mellon's Dana Fellows (faculty from liberal arts colleges who spend a year working on educational computing projects in the College of Humanities & Social Sciences) as well as to university faculty developers and other university units. (CDEC's Associate Director, Bruce Sherwood, is also Associate Director of the Information Technology Center, the IBM-funded center for development of the Andrew Advanced-Function Workstation & Network Project).

CDEC staff teach training workshops in the cT programming language for faculty developers from Carnegie Mellon as well as other universities.

CDEC organized and ran the Authors Guild for Andrew (AGA), a support group for faculty developers using the university's Andrew system software for advanced-function workstations.

CDEC holds regular monthly seminars on its own research and development projects.

CDEC staff ran the 1987-88 Understand Seminar, an interdisciplinary research seminar sponsored by the Psychology Department for the cognitive and computer science community.

CDEC organized a campus-wide seminar series on Intelligent Tutoring Systems.

3. Education & Integration in Curriculum

Integration of computing into the curriculum, educational planning and intellectual life of the campus is a CDEC mission. In addition to the services listed in this and the last section, CDEC leadership works with campus leadership, as exemplified in the following co-authored report:


CDEC supports courses and programs in educational uses and issues of computing. The development of university courses has become a responsibility and point of leadership for CDEC as a result of the quality and academic status of its research staff and the CDEC Director's appointment as Vice Provost for University Studies. CDEC has responsibility for developing and managing the Computer Skills Workshop (CSW), a university core course in 'computer literacy' required of all in-coming undergraduates. Sloan helped support the inception of this course. CSW is the major component of a broader program -- the Computer Languages, Applications and Systems Program (CLASP) -- that sponsors for-credit computing courses that fall outside the mandates of academic departments. (Note: our Department of Computer Science is a research department without an undergraduate program, which undertakes the teaching of computer
languages only as vehicles for instruction in the science of computing. Hence the need for a university unit to undertake responsibility for more general education in computing.) Courses initiated, organized or taught by CDEC under CLASP run the gamut from a required freshman course to graduate seminars taught by Software Engineering Institute staff and include:

- **Computer Skills Workshop** (which has evolved from an introduction to the tools, concepts and utilities of mainframe to personal to distributed computing on both microcomputers and advanced-function workstations; required of all in-coming students; therefore the largest course on campus; CDEC and CSW staff).

- **Introduction to the Andrew System** (CSW staff)

- **Design & Programming in cT** (CDEC staff)

- **Design & Programming in HyperCard** (CDEC staff)

- **Introduction to COBOL** (for applied math and industrial management majors)

- **Instructional Design** (CDEC staff seminar for CDEC Scholars)

- **Design for Computer-Based Learning** (CDEC staff seminar for CDEC Scholars)

- **Formal Methods of Software Engineering** (SEI staff)

- **Software Project Management** (SEI staff)

- **Prolog: Logic & Artificial Intelligence** (CDEC staff)

- **Computer Tools for Problem Solving in Mathematics** (Sloan-supported course)

- **Technology & Literacy: From Clay Tablets to Spreadsheets** (CDEC/AC staff)

- **Computer Technology & Ethics** (CDEC staff)

- **Social Choice Theory & Computer Simulations** (CDEC staff)

- **Social & Ethical Issues in Computing** (CDEC & Academic Computing staff)

The CDEC Scholarship Program for Educational Computing (CSPEC) is another important educational and university service, which CDEC established in 1986 to support educational computing and service projects by qualified students. CDEC Scholars have produced award winning software, taught workshops, made presentations at university conferences and tutored President Cyert in the Andrew system and cT. Several graduates have found computing jobs with Carnegie Mellon and other universities. For details and results of the program, see:

II. Strategic Initiatives & Projects

0. The Center for Design of Educational Computing

The single most important achievement to be credited to the Sloan Foundation grant is the founding and flourishing of the Center for Design of Educational Computing (CDEC) itself.

0.1 Why the Focus on Design?

CDEC focuses on design because design issues are instrumentally and substantively central to both research on and development of educational applications of computing. Design issues are as paramount and pervasive as they are problematic. While our emphasis has been on advances in design and development rather than production per se, CDEC has been very productive.

We believed that a nationally prominent research and development center for design would be the most productive investment for the future of advanced computing in the liberal arts, rather than a decentralized diffusion of educational computing resources across the campus.

The computing revolution for education on campuses has simply not arrived, despite all the hope and hype created by the rapid evolution of the technology itself. Advanced computing technology itself (like Carnegie Mellon's Andrew system) is only the vehicle and utility; its design and engineering for education is a long-term substantive project in its own right. CDEC aims to help lead this effort by providing tools and vision as well as select, strategic applications.

The strategies and future of educational computing must be guided by design -- not forced or occasioned by the technology, its markets or its vendors alone; not entrusted to serendipity and opportunism. There may be more metaphor than method to strategic planning or management by design or designing an organization (like CDEC), but these strategic process and management issues are nonetheless design problems par excellence, design problems of the most genuine and generic kind:

When the mapping of actions on states of the world is problematic, then, and only then, are we faced with genuine problems of design. [Herb Simon, Models of Discovery, p. 162.]

Nowhere is the adaptation of means to ends and the adjustment of priorities among competing ends more problematic than in planning, managing or designing the future of educational computing.

Design is also more than an instrumental or strategic planning issue; it is a rich topical nexus of interdisciplinary perspective, an intellectually substantive concern that informs many CDEC projects. We have capitalized on the importance and focus of this topic on campus. I have edited a volume of papers by Carnegie Mellon faculty from across the arts and sciences; several of the papers in this volume were written by PI's of CDEC- and Sloan-supported projects and motivated by those projects. Design-thinking on all these levels is crucial to the reflective practitioner in educational computing. See:

0.2 Interdisciplinary Infrastructure.

CDEC has assembled a professional interdisciplinary staff of national reputation who represent the full spectrum of liberal arts and each of whom combines strong academic credentials, technical expertise and practical experience in the design, development and deployment of educational applications as well as research interests in disciplinary methodologies, instructional design, cognitive science or learning theory. CDEC research and administrative staff all have joint appointments or teach in an academic department of the university. CDEC is also actively concerned with promoting the educational benefits of computing beyond the groves of academe, for the general public and secondary education.

0.3 Why an R&D Center?

Why should resources for educational computing be committed to or disbursed through a centralized agency, in particular an R&D center rather than a pure service organization?

One would not expect leadership in, say, the field of statistics with a handful of statisticians disbursed among the myriad departments requiring statistical tools and analysis; leadership in any field requires some concentrated 'critical mass' of applied research activity to track and advance the state-of-the-art. Educational computing, on CDEC's model, is not merely a distributed utility or service like electric power, statistical analysis or academic computing support; it is an area of applied science and art that draws on many disciplines and leverages leadership for the greatest impact in the long term. Leadership in the large -- a leadership role for any university in the world at large over the long term -- requires more than diversified, local innovations across its constituent disciplines.

Like any center, CDEC collects, develops and capitalizes upon the best expertise available to build strategically for the future, trading off short-term against long-term gains (150 seed projects today that cannot be sustained across advancing technologies against the 50 that will propagate impact long into the future). This requires strategic planning, research as well as development, intensive focus and interdisciplinary exchange, continuity over time for tracking changing technologies, a critical mass of expertise, and a stable base for leveraging available resources and opportunity into striking innovation and expanded productivity -- the *raison d'être* 's for any dedicated center, institute or department.

Only a centralized infrastructure could develop the likes of cT, the nexus of research and development expertise we have in intelligent tutoring systems, or the multi-media initiatives described below. Only a centralized infrastructure could coordinate and leverage the mass of outreach and educational services CDEC provides on such a wide front.

0.4 CDEC's Mission & Measure

CDEC's mission is to provide leadership and support for the design and development of advanced computer applications for the liberal arts. This report provides some measures of CDEC's success. CDEC aims to generate strategic planning, design wisdom, dedicated academic and technical expertise focused on the problems and promise of advancing technology for the liberal arts. But CDEC has also distributed both substantial funding and equipment grants across campus with strategic and seminal impact in mind (see *CDEC Productivity Report 1984-88*, CDEC
Neither CDEC (nor any *de*-centralized effort), even at several times the cost, can populate the liberal arts with sophisticated educational applications of computing. Saturation will come only by evolution over time -- and only by putting good tools, models and know-how in the hands of faculty so that they can then produce software and applications as naturally as they now produce books, articles and lectures. Besides good tools, faculty efforts and the watching world also need good standards and trail-blazing models. CDEC has concentrated on developing the tools, showcase models and know-how needed to achieve excellence as well as wide use and impact over the long term.

CDEC is an R&D center, not a Volkswagen factory or a welfare agency. We do not dole out support to any person who has a bright idea for a computing vehicle for education. We do not build educational computing vehicles for the sake of generating the greatest possible number of visible applications on campus. We do not limit support to pedestrian vehicles that can proliferate in great numbers. Instead, we look ahead, we track the states of the relevant arts, we assess needs and comparative advantage, and we concentrate our resources in select areas for the greatest strategic gain, by the lights of our accumulated experience and expertise, the likes of which are not plentiful among the faculty at large. Our strategic vision is based on state-of-the-art knowledge of both computing and education across many domains not-available to parochial faculty.

For all its selectivity, CDEC has nonetheless spawned and supported 75% of the educational computing applications developed on campus and 95% of the Andrew applications of the last four years: (See *CDEC Productivity Report 1984-88*, CDEC Report # 87-17). Technology transfer, dissemination, faculty support and direct educational services have been important to CDEC’s mission. But the crucial factors have been our focus and selectivity, the way we have built expertise and concentrated resources over time in certain strategic areas and seminal projects: there is a long-term strategy at work here that would not emerge from a less dedicated, less focused, less interdisciplinary, less venturesome enterprise. CDEC’s activities encompass three categories. This part of the report will focus on area 1, our R&D initiatives and projects.

1. Research & Development
   1.1. Productivity Tools
   1.2. Model Software
   1.3. Multi-Media Environments

2. Technology Transfer & Dissemination

3. Education & Integration in the Curriculum.

0.4 CDEC Criteria for R&D Projects

Tracking the development of high-end microcomputers, advanced workstations and distributed computing technology, CDEC’s principal mission has been to research the design and support the
development of advanced applications in the liberal arts. CDEC has initiated and supported educational computing projects that promise to be both strategic and seminal.

A **strategic** application is one that addresses pandemic educational problems or crucial bottlenecks, promotes intellectual skills not well promoted by traditional means, effects improvements in the learning process *with* the computer that cannot be accomplished *without* the computer, and forces us carefully, critically and creatively to rethink, reaffirm or revise our basic concepts, pedagogy and goals.

A **seminal** application is one that will *propagate* use, impact and benefits in more than one domain or discipline, serve as a paradigm for propagating needed innovation in educational strategy or curriculum, and exploit or advance important disciplinary or educational research.

These criteria also ensure that our applications are intellectually substantive, academically rigorous, and well integrated into educational and research agendas of the university community, motivating or responding appropriately to needs for curricular and pedagogical reform.

An example of an application that is both strategic and seminal is **PD World**, a simulation environment (built in cT) for generating and testing Prisoner-Dilemma-type models of the evolution of cooperative behavior and collective rationality. **PD World** is strategic because it addresses an analytical approach that is hard to teach in general educational settings but that is crucial to the understanding of important work in several disciplines. It puts serious research tools in the hands of students as only the computer can: the environment is extensible, allowing a student to either replicate or extend all extant research (such as is described in Robert Axelrod's seminal book, *The Evolution of Cooperative Behavior*). **PD World** is seminal because it invites and propagates use across a spectrum of disciplines: philosophy, political science, decision science, economics, social psychology, policy analysis, socio-biology. These features have made it integral to courses in the Humanities & Social Sciences College, the Graduate School for Industrial Administration, and the School for Public & Urban Affairs: the intellectual substance of the program is integral to major research and problems for social institutions and public affairs.

As further assurance that CDEC's thinking and applications be informed by and integral to the intellectual life and curricula of the university, all our research staff have joint appointments or teach in academic departments and CDEC's Director serves as Vice Provost for University Studies, overseeing general education initiatives and curricular reform across the university.

Because the discretionary Sloan grant has made strategic planning possible and because CDEC's strategic vision is itself an important outcome, I have organized this part of the report according to the categories of CDEC's strategic R&D initiatives:

1. **Productivity Tools:** Solving Problems for Developers
2. **Model Software:** Trail Blazing from CAI to AI & ICAI
3. **Multi-Media:** New Media for Values Education.
1. **Productivity Tools: Solving Problems for Faculty Developers**

CDEC addresses one strategic need in the development of educational software: better, more sophisticated means of production for faculty and students in modern computing environments. Major achievements have been Common Lisp and AI tools for Carnegie Mellon's Andrew system and student workstations, multi-media production tools, and the cT programming environment, which is portable across diverse operating systems and computers.

1.1 **Common Lisp & AI Tools**

CDEC tailored a full-function Common Lisp and other artificial intelligence (AI) development tools (e.g., OPS5) to Carnegie Mellon's Andrew system and Unix workstations. Both public domain Kyoto Common Lisp (KCL) and the enhanced and commercially supported Ibuki Common Lisp (IBCL) have been made available to support AI applications and courses on our advanced-function workstations. KCL/IBCL were chosen because of reasonably sized core images, in order to run under four megabytes (the typical memory on workstations commonly available to students). Functionally equivalent or superior Common Lisp environments require twice or more the memory as KCL/IBCL. Further specifications on the technical environment are found in:


Among the important applications exploiting the Common Lisp are the CMU Proof Tutor and the VALID on-line logic course described below. In addition, several undergraduate and graduate courses -- in philosophy, psychology, social science, mathematics and computer science -- use the Common Lisp environments, from introductory computing to a course in *Interactive Fiction*.

1.2 **Multi-Media Tools**

The strategic importance of computer-based multi-media is discussed at length below. Two CDEC projects are aimed at providing better tools and environments for this technology. One is the development of the *VideoDisc Controller (VDC)*. The VDC, software presently implemented on the Macintosh and to be ported to the advanced-function Unix workstations, enables precision frame-grabbing off multiple videodiscs running on multiple players. This provides a working model for accessing video archives off multiple devices on a distributed network, allows video material from several discs to be edited and programmed together for rapid prototyping of video sequences, and is adaptable to classroom presentations where one needs access to multiple videodiscs (e.g., for studying films such as Citizen Kane, which occupies four discs).

Another is a model *multi-media workstation*, which exploits an advanced-function Unix workstation, MIT's Video-X and X-Windows software, and a Paralax™ video-digitizing board for running real-time video from any analogue sources in a moveable, sizable window along side other windows and processes on the screen.
CDEC is adapting cT as a video authoring and control language in the process of porting its videodisc control programs (for the projects described below, developed on IBM-based systems) to the Macintosh and Unix workstation. This will provide a development and delivery language for videodisc programs that will run on the major microcomputers as well as Unix workstations, thereby obviating one disincentive to the development of videodisc applications: portability across systems and the consequently meager installed base equipped to run such programs.

CDEC has also built video editing and X-function extensions to HyperCard and provided video equipment for faculty and students working on interactive access programs for generic videodiscs in foreign language instruction. See also:


1.3 The cT Programming Environment

CDEC's premier 'shrink-wrapped' product for distribution is the cT programming language and environment for the Macintosh. (A copy of the Macintosh version is included with this report. IBM's Academic Information Systems is distributing CMU Tutor, the prototype of cT developed for Unix workstations. CDEC itself is distributing all versions of cT.)

To address a basic productivity problem, how to get faculty eventually to generate a broad, rich applications base, we have addressed the heart of the problem: the need for better, easier, more sophisticated productivity tools for implementing innovations in increasingly sophisticated environments. The cT programming language is the prime analogue (although it is not the only example). CT is easy to learn, easy to use -- relative to any extant general programming language; it makes it easy to control the modern interface features of advanced computers (which make modern programming all the harder for novices and experts alike); and it is machine and operating-system independent, allowing applications to run unaltered on the most popular microcomputers as well as advanced Unix workstations, thus obviating the hardware incompatibility problems that are a major disincentive to faculty developers.

CT was designed to address three major obstacles associated with conventional programming languages that inhibit faculty (or student) developers of educational software:

1. Difficulties or inefficiencies in learning and use.
2. Lack of support for modern graphical interfaces.
3. Lack of portability or compatibility across hardware and operating systems.

1.3.1 Development Problems Addressed by cT

Despite 20 years of hope and hype, computers have not yet had the expected profound and widespread impact on education at any level -- the computer 'revolution' for education is disappointingly late in arrival.

It is useful to distinguish generic academic tools and applications (like word processors, spreadsheets, database management systems, etc.) -- whose development and impact have proliferated -- from substantive educational applications designed to aid learning or instruction in specific academic subjects -- whose development and impact have not. There is plenty of market incentive for commercial software houses to develop the former: the applications are generic, their market is broad, their design problems are relatively independent of specific intellectual content or skills, and their utility and adoption is unimpeded by academic theologies or skepticism.

Educational applications, on the other hand, require subject matter expertise and pose the disincentive of much more limited markets. Applications targeted at general education (non specialist) audiences in a wide variety of settings -- audiences who are unpredictably mixed regarding interest, knowledge or skills -- impose special design and development challenges: whereas the mechanics, niceties and functionalities of word processing or spreadsheet systems are
generic and relatively straightforward, the intellectual operations we wish to illuminate for non-specialists in substantive domains of the arts and sciences are not so easily amenable to computer interaction. Designing computer-based interactive experiences for mixed audiences is a ticklish, time consuming task. Yet the demonstrable power of the computer in so many other areas of human enterprise beguiles us with the promise and challenge of realizing similar value for education.

The development of any computer application requires at least three types of cyclical effort: (1) substantive, pedagogical and design research, (2) iterative design (through cycles of incremental prototyping and formative evaluation) and (3) implementation, i.e. programming. The intellectual, pedagogical and design issues endemic to developing good graphic interfaces and effective interactions for educational impact require considerable effort on the part of subject area experts, quite apart from the onerous programming effort. But the research and design efforts are at least tasks that faculty are prepared to undertake and that reward effort with better understanding of one's educational goals, pedagogy and even one's field of work. Programming, for iterative design and implementation, is another matter. The following are three major problems impeding the development of good educational applications:

Problem 1: Programming costs to faculty in time, effort or money are too high. A major difficulty and disincentive posed for faculty developers by conventional development environments lies in the programming effort itself. Good educational programs require large amounts of well designed graphics and interaction with the student user. Until recently, both have been notoriously difficult and labor-intensive to program 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. Conventional programming languages are neither easy to learn nor to use, especially for the design and control of modern, graphical user interfaces.

Problem 2: Conventional programming languages are very inefficient for either design or production of modern user interfaces, even by experts. Worse, they provide little specific support for the crucial graphic interface and interaction required of good educational applications. Even if the faculty member is willing to spend the time and effort to program herself (or the money to hire expert programming talent), these commodities are spent inefficiently in conventional programming environments, which make the crucial graphic and interactive design process more difficult than necessary.

This inefficiency is compounded by the need for iterative design through rapid cycles of incremental implementation and formative evaluation. One does not simply design an application on the proverbial 'drawing board' and then turn it over for coding: programming and formative implementation are required very early in the research and design cycle. Iterative design and incremental implementation must be quick and efficient, whoever does the programming.

The inefficiencies of conventional programming environments for developing any application thus does two-fold harm to the development of educational applications: (1) Faculty (unlike commercial developers of generic academic software with large markets) cannot afford the time or money; estimates range as high as 1000 hours of development time for one hour of interaction time or instruction. Consequently, (2) too many educational applications are badly designed and give the enterprise a bad reputation. It is crucial to realize that inefficient programming environments not only make for low productivity but also for bad design, because they discourage or cut short the iterative design-implementation-evaluation cycle essential to good design.
Problem 3: Compatibility and portability. Suppose that a faculty member can afford to spend the time or money required and, further (after great investment and significant risk to her career) actually develops a good, effective, well designed educational application. The next problem confronting the enterprise is wide deployment and dissemination -- for either adequate testing and peer review or widespread use and impact. Conventional programming tools have been tied to particular machines; that is, a given application can usually be developed on only one kind of machine and therefore will run on only one kind of machine: software developed in conventional programming languages on one kind of hardware will not be compatible with other types. A separate and costly effort must be made to port software for, say, an IBM PC to, say, a Macintosh. The centrally important graphics and interaction for educational applications must largely be reimplemented to make the program run on any other machine type.

The result is that, 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 or substantive education. And those educational applications that happen to be developed are too often badly designed.

Few professors have either the programming skills or the time to develop educational programs. When they do develop such programs and those programs are good and useful, they cannot easily share them with colleagues or disseminate them widely because of their dependence on a particular type of computer (e.g., Apple II, IBM PC) and operating system (e.g., Apple OS, DOS).

The advent of yet more powerful personal computers (the Macintosh II, IBM PS2 series, and the Unix Workstation) will, if anything, make these problems worse. While application programs on these machines can be elegant and easy to use, building such programs is a task only for the expert: it requires not only experience in yet more difficult modern programming languages (e.g., "C", object-oriented Pascal), but also mastery of all the new special techniques for controlling modern interface features such as pull-down or pop-up menus, multiple windows, multi-tasking, special fonts, cursors, and mice. Such features are even less portable across computer types and operating systems than their cruder predecessors.

Any revolution for educational computing requires that faculty acquire better means of production.

1.3.2 How cT Solves these Problems

The cT programming and authoring language has been designed to solve these technological problems (cost and difficulty, inefficiency, lack of compatibility) that deter good design and efficient, affordable development of educational applications. (The cT language is also well adapted to creating research tools, classroom aids, and other generic academic applications.) Specifically, cT embodies the following desirable features for educational developers:

1. CT is a programming language that is easily learned and readily usable by non-experts. For example, faculty from a variety of colleges and universities, with little prior programming experience, have learned cT and produced a significant piece of instructional software during one-week workshops.

Others have acquired the basics with a few hours of introduction, and gone on to use the rich on-line documentation and help systems to master the language and produce software on their own.
Features making cT easy to learn and use include:

(a) **On-line, dynamic 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 programmer can alter this sample code to see and 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.

2. **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 mouse input, intermingling of multi-font text and graphics.

3. 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 the Sun, MicroVax and IBM RT 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 (Macintosh OS, DOS, Unix, etc.) so that use of cT does not interfere with other work.

4. 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 Fortran), cT has hooks to other languages and can call packages like 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.

5. **cT** is a full 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. For example, cT has been used to construct an "intelligent tutoring system" called Sketch (for sketching algebraic equations) involving state-of-the-art techniques from computer and cognitive science, which won
Advanced Computer Applications for the Liberal Arts

an EDUCOM/NCRPTAL Distinguished Software Award in 1987.

While enhancements will continue indefinitely, cT already provides a machine-independent, non-experts' educational programming environment powerful enough for experts as well.

In sum, CT offers the following unusual strengths for modern computing environments:

* incremental compiling
* interactive graphics in windowed environments
* instant portability across diverse computers
* automatic rescaling of text and graphics
* multi-font text
* menus (which operate appropriate to the computer, in pop-up or pull-down mode)
* mouse and keyboard inputs
* analysis of words and sentences
* analysis of numbers and algebraic expressions
* rich sequencing options
* standard calculational capabilities
* numeric and text files
* on-line reference manual with executable examples
* accurate and informative error diagnostics

CDEC will further enhance cT as an environment for developing interactive video control programs and interfaces.

1.3.3 Availability and Use of cT

CT is currently available for the Macintosh with on-line documentation, textbook and user manual (as of July, 1988). Versions for the IBM PS/2 series and the DEC Vax, Sun III and IBM RT Unix workstations will follow during 1988-89. A videotape is also available that provides an illustrated introduction to the language and what can be done with it.

An important application of cT is to develop sophisticated modern interfaces for educational or research software built in other languages like Lisp or C. Links to Fortran will exist soon as well. For example, this allows faculty to build AI engines in Lisp but construct user interfaces more readily in cT, as has been done with the CMU Proof Tutor (below), or to capitalize on powerful research programs for student use by building user-friendly instructional interfaces.

CT has 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). One important strategy for encouraging widespread faculty participation in the development of substantive educational applications of computing is to capitalize on the computer as a research tool and the potential educational use of research programs in upper division curricula. New development tools like cT make it expedient for faculty to build user-friendly interfaces, interactive instructional modules, graphic data display, analysis and plotting programs or other educational tools on top of powerful state-of-the-art research programs.

Faculty also use cT for easy production of classroom interactions and presentations. See:
CT is, of course, typically used to construct sophisticated applications de novo. For example:

*Graphs & Tracks*, a simulation environment and intelligent tutor for introductory kinematics that won two EDUCOM/NCRIPTAL awards in 1988 ("Best Physics Software" and "Best Integrated Software").

*Sketch*, an "intelligent" tutor for graphing algebraic equations and the largest program written in cT, won a 1987 EDUCOM/NCRIPTAL Distinguished Educational Software Award. Other major educational applications built with cT are in use in a variety of disciplines and institutions.

*PD World*, a simulation environment built in cT for developing Prisoner-Dilemma-type models of the evolution of cooperative behavior, allows the replication or extension of all extant research in the field of social choice theory (thus putting a serious research tool in the hands of students as well as faculty). *PD World* is used in courses in philosophy, political science, decision science, economics, social psychology and socio-biology.

A diskette sampler for the Macintosh is available from CDEC containing a whole suite of programs written by physics faculty in cT.

CT has been adopted by myriad faculty across the spectrum of the liberal arts, at Carnegie Mellon and other schools, for all these purposes and has become a cornerstone of Carnegie Mellon's long-term strategy for facilitating wider development of educational applications by faculty.

CT has also been used by students for the development of original educational software. Three stellar examples actually developed for use in undergraduate courses are *Chemistry Lab* by David Thompson, *Genetics Lab* by Pam Reinagle, and *Nuclear Power Plant Simulator* by Stuart Shapiro. Shapiro's project was in fact saved from fatal implementation difficulties in the C programming language by the decision to design and implement in cT. (All three of these students were CDEC Scholars: see the CDEC Scholarship Program in Educational Computing, under Education, above.)

Two dozen sample cT programs have been distributed by IBM and with Carnegie Mellon's Andrew system as well as on the ICEC-Ware tape (disseminated among the member schools of the InterUniversity Consortium for Educational Computing).

CDEC offers faculty and high school teacher workshops in cT as well as for-credit courses for Carnegie Mellon staff and students. Workshops have also been given at Carnegie Mellon, Vassar College and Cal State, Northridge for members of the InterUniversity Consortium for Educational Computing. Developer workshops run to date (under the sponsorship of the InterUniversity Consortium for Educational Computing - ICEC) include Carnegie Mellon (1986) (100 participants), Vassar College (1988) (50 participants) and California State College at NorthRidge (50 participants).

CT puts sophisticated means of production in the hands of faculty and students, eliminating or lessening reliance on expensive applications programmers. Whatever other forms of material support are provided faculty, sophisticated, easy-to-master productivity tools must be provided that insulate developers and end-users from the vagaries and incompatibilities of advancing technology. CT will cut costs of development on the part of novice and expert developers alike.
whether those costs are born by individuals, funding agencies or universities.

The pay-off from cT is truly strategic: only tools like cT will eventually make possible wide development of thousands of applications able to run on a variety of systems, rather than a few major applications (for the same investment). Without this kind of generator and tool, the day when a 'thousand flowers bloom' will never come. CT also makes possible more economical development of major showcase applications, many of which will remain cost-intensive nonetheless but less so than in the past.

1.3.4 Historical Note

The cT language is the "grandchild" of the Tutor language (1967) developed as part of the University of Illinois PLATO project. CT's emphasis on interaction and graphics, and it's design for non-expert programmers reflect much in the original Tutor language. MicroTutor (1977) was designed and implemented to implement appropriate aspects of Tutor on micro-computers. CT, first known under its prototype name "CMU Tutor," retains many features of the earlier languages in a much more powerful version, providing both much extended support for the user and full control of modern machine interfaces, as well as machine independence. CT was developed by Bruce and Judith Sherwood, David Andersen and Kevin Whitley at the Center for Design of Educational Computing at Carnegie Mellon University. The Sherwoods and Andersen were also principals of the team that developed Tutor and MicroTutor.

1.3.5 References

The following articles contain reviews of cT (under its prototype name, CMU Tutor) and place it in the larger picture of development tools and strategies for educational software:


Clayton Lewis and Gary M. Olson, Can Principles of Cognition Lower the Barriers to Programming? Paper delivered at a workshop held at the University of Colorado, July 1986 (available from the authors, respectively, at the University of Michigan, Ann Arbor, and the University of Colorado, Boulder).

The story of cT, historically and technically, and its exemplary applications is of interest in its own right. Besides the cT book and manual, the following papers and articles chronicle the development and strategic impact of cT:


Covey, Preston K., Joseph Devine, Stephen Fienberg and Christine Neuwirth. The Integration of Educational Computing in the College of Humanities & Social Sciences. CDEC Technical Report # 88-17.


2. Model Software: Trail Blazing from CAI to AI & ICAI

Besides good productivity tools, the field of educational computing needs good model applications that set standards and exemplify what it means to be strategic and seminal in the use of advanced computer technology.

Our applications attempt to track and advance the evolution of strategically important types of application: sophisticated simulation, graphics and data-analysis environments and 'intelligent' computer-assisted instruction (ICAI) and tutoring systems (ITS). We have avoided projects that 'reinvent the wheel,' focusing on projects that promise a unique or distinctive contribution to their fields and that capitalize on or create comparative advantage at Carnegie Mellon.

Carnegie Mellon has distinctive strengths in cognitive and computer science across the arts and sciences; it is natural then for CDEC to incorporate the application of cognitive science, learning theory, artificial intelligence and expert system research to education.

CDEC educational software projects are intimately connected with on-going research agendas in their respective fields and special emphasis is placed on design research and modern interface design. There are generic design and implementation lessons to be learned from our work that should help propogate research, design and development strategies beyond the shelf-life of any particular application. These lessons and our research agendas are reflected in our technical reports listed above in Part I section 1.4 Research Agendas.

The projects catalogued below exclude the many CDEC-supported projects in the engineering, business and public policy schools. Through faculty competition for grants which CDEC administered, CDEC has supported over 150 educational software projects across campus since 1984-85. (See CDEC's Educational Software Catalog, May 1986.) Sloan funding has supported a third of these. (Projects/programs are difficult to count, since some projects/instructional packages consist of multiple programs that are not mere 'routines.' ) Sloan funding has had double value in that every Sloan dollar has saved a CDEC dollar towards another project; the whole impact of Sloan funding is therefore larger than the sum of its parts.
Sloan-supported projects are indicated by asterix and catalogued with other CDEC projects according to the colleges and departments in which they were developed. Multi-media applications are described separately in the following section, since this agenda has strategic significance in its own right.

2.0 The Carnegie Mellon Andrew Project

Many of the projects below are described as running on "the Andrew system" or "advanced-function workstations." The latter are powerful Unix-based workstations of the; the former denotes two things: both system software that runs on top of 4.2/4.3 BSD Unix and the distributed campus-wide network and file system. The bi-valent Andrew project as a whole was named for the two Andrews, Carnegie and Mellon. Many of the projects below aimed to exploit the multi-windowing and processing capabilities of the workstations and the facilities of the distributed file system and campus network (e.g., the Comments program for collaborative writing and learning). The Andrew project, a joint IBM/Carnegie Mellon venture, is described within various comparative contexts in the following:


2.1 The Fine Arts

We include dimensions of the fine arts or design within the scope of the liberal arts and have supported projects that address higher-order reasoning and design skills and that exploit sophisticated graphics, calculation or expert-system technology.

2.1.1 Architecture

* PRINCE: A Three Dimensional Graphics Program. PRINCE (PRototype INteractive Creative Environment) was designed by Gerhardt Schmitt to satisfy a Prince, and it would:

'DRAW ME A HOUSE PLEASE' the Little Prince said to the Architect. [St. Exupery]
Developed in C for the distributed Andrew computing environment and advanced-function workstation, PRINCE allows the sketching of ideas and the transformation of those sketches into three dimensional graphic representations as well as the interactive creation of fractal images.


Structural Systems Design Tutor. This project was begun by Professor Mary Lou Maher in Civil Engineering and taken over by Professor Irving Oppenheim in Architecture.

This program was developed to help architecture students understand structural engineering. It provides an environment for synthesizing structural systems, using several windows on the screen: a working window for combining structural components, a substructure window for defining new components, a plane window for displaying any 2D plane in the building, and a window for viewing the 3D display of the structure. The program analyzes and judges the student's proposed structure for overall feasibility, once the student chooses specific structural elements and arrangements from a database of building materials.

The tutor environment consists of C and Fortran programs now running on the Andrew system and advanced-function workstation, having begun life on the IBM XT. Professor Oppenheim's project was to port the environment to Andrew and expand its facilities and uses. The software provides a graphics-based expert-system environment to aid students in the study and manipulation of the behavior of structural systems. The environment's calculation and graphics facilities synthesize and display structural data in three dimensions; its expert system generates and solves problems on statically determinate structures in three dimensions. The environment includes testing and grading facilities. See:


2.1.2 Art

*PANDA* (Painting and Drawing Aid) was developed by Professor Harry Holland. PANDA was begun on the IBM XT in monochrome and has evolved to the IBM AT with a full 512 color palette. It is an extraordinarily sophisticated and dramatic electronic painting and design environment for studying theory and composition. It is the kind of spectacular application that has received national press in *Time* Magazine and has allowed a diversity of students from the humanities and sciences to participate in Professor Holland's electronic studio courses in our Fine Arts College, where ordinary studio courses are closed or too advanced for non-majors. In this way, PANDA has helped the fine arts to fulfill their liberal arts mission. Sloan provided both hardware and support for the development of professionally designed and tested documentation for this very complex software system, through the Center for Document Design.
2.1.3 Music

CAMUS (Computer Aided Music) and Solfege are interactive, graphic drill programs developed for IBM and Apple microcomputers by Collette Wilkins and now published by CONDUIT at the University of Iowa.

* VOICE is an intelligent music theory and vocal composition tutor developed by Professor Marilyn Thomas. Versions exist in Pascal for the IBM PC and in C for the Macintosh and the advanced-function Andrew workstations equipped with a Yamaha synthesizer. The Macintosh version is called MacVOICE. VIVACE is another project of Professor Thomas's, a rule-based artificial intelligence system for composition, written in Lisp for mainframe or mini computers equipped with a Yamaha synthesizer. See:


* Pluck is a simulation/animation environment with synthesizer developed by Professor Ned Vander Ven (Physics) in cT for studying the physics of musical acoustics through the manipulation of 'violin strings.'

* The Music Box & Musical Dictionary. This and other software in cT employing a computer-controlled synthesizer for storing and manipulating musical examples digitally and providing analogue output is under development by Professors Vander Ven and Robert Schumacher (Physics) for the Musical Acoustics course.

2.2 The Natural Sciences & Mathematics

2.2.1 Biology

* Cellular Growth Simulation. Professor Linda Kauffman (Biology) and Judith Sherwood (Senior Developer Consultant, CDEC) developed simulation software for studying the growth and fission of bacterial and viral cells and the dynamics of bacterial-viral populations. Learning to obtain experimental data on cell population growth and to plot, analyze and interpret such data are basic skills required of experimental biologists in microbiology, molecular biology and genetics. The simulation software includes two modules, on simple bacterial growth and on viral reproduction, with calculational and graphing facilities. (IBM PC and Andrew system workstations.)

Genetics Engineering Lab. CDEC Scholar Pamela Reinagel developed a simulation environment, written in cT, for genetic engineering experiments. The program allows rapid prototyping and testing of experimental designs, so the student can both see the outcomes of a large number of designs and also make more efficient use of 'wet lab' time conducting live experiments.
Thus, the program is a complement to actual lab work. It also allows the student to supplement lab work by designing and simulating experiments that could not be done within the available lab time or facilities. (Macintosh and Andrew workstations.)

2.2.2 Chemistry

* Density Matrix 3-D Graphics System. Professor Aksel Bothner-By developed a 3-D graphics system for tutorial instruction in and execution of density matrix methods, with special applications for the analysis of nuclear multi-pulse NMR spectra. This area is exemplary of an important application of computer graphics: where the results of calculation are not intuitively obvious and graphics provide surprising insights. The system performs calculations and presents the results in graphical form; it includes tutorials, but the calculational and graphics facilities can be used alone as a workbench for research. (IBM PC/XT with color monitor and Enhanced Display board.)

* Chemical Equation Solver. Professor Paul Karol developed a software package for guiding students through complex calculations in chemical stoichiometry, to help students integrate the mathematical and chemical knowledge required to balance chemical equations in equilibrium problems and avoid losing sight of the basic intuitive chemistry amidst the complexities of the mathematics. The first year/phase of the project (1984-85) developed the syntax and semantic procedures for entering chemical formulae, implementing the mathematics for balancing equations, and developing tutorial mechanisms for guiding students through the procedures and problems. The second phase/year of the project (1985-86) ported the software from mainframe BASIC to Turbo Pascal for the IBM PC. After designing and developing the original model with Professor Karol, CDEC Scholar David Thompson then reimplemented the chemical equation solver with an improved interface in cT for the advanced-function workstation and Macintosh.

Chemical Drawing Board. CDEC Scholar Christopher Lewis developed an interactive 'drawing board' or 'palette' for constructing and checking correct chemical structures, allowing that no two people will draw the same molecule in exactly the same way. Written in cT for the Macintosh and advanced-function workstation, the program is a generic tool for use across the natural sciences -- biology, chemistry, biochemistry, chemical engineering etc.

* Physical Chemistry Simulation Software. Professor Gary Patterson developed 2-D graphic and calculational programs in cT for the Andrew workstation for simulating and calculating molecular structure and behavior using the tools of chemical kinetics, statistical mechanics, and quantum mechanics.

* General Chemistry Simulation Software. Professor Robert Richman, with the collaboration of other chemistry faculty, developed a set of kinetics software for teaching the relation of macroscopic results to the microscopic world in an intuitively perspicuous but mathematically sophisticated way. Written in C for the advanced-function Andrew workstations, the software exploits the multiple window and processing capabilities of the workstation by showing experimental parameters in one window, explanations and calculations in another window, and graphing the results in a third window. Students can work in novice mode, which guides them step-by-step through a series of tutorials, graphics and problems, or expert mode, where they construct problems and model the behavior of systems of their own choosing. The software has been used in Intro Chemistry, Dynamics and Equilibria, and Organic Chemistry.
* The LHASA program in retrosynthetic analysis, developed in Fortran at Harvard as a research tool and modified at Tufts University, was imported and further modified by Professor Richman's team for use with undergraduates in general, organic and experimental chemistry courses. (VAX and MicroVAX II systems.)

2.2.3 Physics

* Graphs & Tracks: Kinematics Tutor & Simulation. CDEC Research Scientist David Trowbridge developed this kinematics tutor and simulation environment, which won two awards in the 1988 EDUCOM / NCRPTAL Higher Education Software Program: "Best Physics Software" and "Best Integrated Software," for the integration of student guidance with student-controlled construction. The program, written in cT for the Macintosh or advanced-function workstation and designed to teach kinematics concepts and graphing skills, supports two activities critical to introductory kinematics: "From Graphs to Motion" presents graphs of position, velocity and acceleration versus time. Students then construct tracks on which a rolling ball executes the motion represented in the graphs. Help is available that provides step-by-step guidance through the solution. Students can generate new problems by creating and storing their own graphs. "From Motion to Graphs" demonstrates the motion of the rolling ball and students must sketch the corresponding graphs. The student is given a graphing palette or may draw freehand. The program provides feedback, focusing on dubious solutions and prompting the student without directly correcting errors. See:


* Optics Bench. Developed by Bruce Sherwood in cT. The student selects among a collection of optical elements such as convex and concave lenses, mirrors and apertures, and places them along an optics bench. A screen is also positioned as desired. Using a mouse to point, the student indicates an origin for a spray of light waves which then pass through the elements. An image forms on the screen and is displayed as a circle shaded according to the intensity of incident light.

Other similar simulation environments developed by Sherwood and colleagues in the suite of cT physics software include:

* Quantum Well.
* Orbits.
* Slope.

* Quantum Well was upgraded by Professor Bradley Kiester for the Wave Mechanics course.

* Fourier Series Tutorials & Simulations were developed by Kiester in cT for plotting functions and solving differential equation problems in the Mathematical Models of Physics course. The programs can be used by instructors for class demonstrations and by students for experimentation and problem solving.

* Equipotential Lines was developed in cT by David Trowbridge (CDEC). The student constructs an arbitrary arrangement of charged conductors having various fixed electric potentials. The program computes values of the potential in the intervening regions and draws lines of constant potential.
Expressions for Waves was developed in cT by David Trowbridge (CDEC). The student is engaged in a dialogue on concepts of waves, including ideas of amplitude, frequency and wavelength. The student is asked to enter simple mathematical expressions that describe oscillations. The goal is for the student to develop skills in writing expressions for sinusoidal traveling waves. At the end of the dialogue, a brief quiz is given. This involves a problem generator: problems are stated in words, with numerical parameters generated at random.

Heat Transfer Tutor was developed by Professor Hani Henein (Materials Sciences) for students in physics, materials sciences, chemistry and chemical engineering; begun on Hewlett-Packard workstations and ported in cT to Andrew workstations. The basic goal is to develop skills in translating qualitative understanding of word problems into quantitative statements and solutions in transient heat flow and lumped capacitance. The tutor presents problems, prompts and guides step-wise analysis of assumptions, initial and boundary conditions, indicating errors but not providing correct solutions. See:


OWL is an Optics Workbench Laboratory developed by Professors Robert Schumacher and Ned Vander Ven (Physics), expanding on the work of Professor Bruce Sherwood's Optics Bench, allowing the user to vary multiple parameters of an optics system without being limited by point source, with calculational facilities for matrix methods and ray-tracing.

Pluck is a simulation/animation environment with synthesizer developed by Professor Ned Vander Ven (Physics) in cT for studying the physics of musical acoustics through the manipulation of 'violin strings.'

The Music Box & Musical Dictionary. This and other software in cT employing a computer-controlled synthesizer for storing and manipulating musical examples digitally and providing analogue output is under development by Professors Vander Ven and Robert Schumacher (Physics) for the Musical Acoustics course.

Dr. Thevenin is a paradigm intelligent tutoring system using expert system technology to address a major bottleneck in science and engineering instruction in the understanding and design of electric circuit systems, Thevenin equivalents. Developed by Sarosh Talukdar and Ros Joobani of the Engineering Design Research Center. Given a moderately complex circuit consisting of a current source, a voltage source, an arrangement of resistors and two terminal points, the student is tasked with finding the Thevenin equivalent from the viewpoint of the two terminals.

The interest to educational computing and the reason for CDEC support of this project are the components of the tutor that make it a 'compleat' exemplar of state-of-the-art intelligent tutoring: A Problem Generator, capable of producing essentially endless streams of problems to fit specifications set by the student. A Problem Solver capable of solving all problems the generator can produce with all the methods the student must learn. An Interface that allows easy and natural communications between the student and computer tutor. A Monitor capable of tracking a student's attempts to solve a problem, identifying any errors, assessing their severity and assigning a numerical grade to the student's efforts. A Teacher who will be responsible for developing models of the student, diagnosing student needs based on performance, devising solution strategies and making explanations of errors. A Supervisor to coordinate the other components.
Joobani, R. and S.N. Talukdar. An Expert System for Understanding Expressions from 
Electric Circuit Analysis. Proceedings of the 9th International Joint Conference on 
Artificial Intelligence, Los Angeles, August 1985.

Joobani, R. and S.N. Talukdar. A Knowledge-Based Intelligent Tutor for Thevenin 
Equivalents. Proceedings of the University AEP Conference, IBM ACIS, Alexandria VA, 
June 1985.

2.2.4 Mathematics

* Sketch: Algebraic Equation Graphing Tutor. Developed by CDEC Research Scientists 
Jill Larkin, Carol Schectic and David Trowbridge, Sketch won a "Distinguished Software" award in 
the 1987 EDUCOM / NCRITICAL Higher Education Software Program. Sketch, written in cT for the 
Macintosh or advanced-function workstation and designed to develop skills in visualizing and 
sketching graphs of algebraic equations, teaches a systematic approach to sketching curves that 
emphasizes a step-by-step procedure for transforming simple expressions into more complex 
one and transforming the graph accordingly, rather than the plotting of individual points. Sketch 
incorporates the following functions and attributes of an artificially intelligent tutor: a model for 
problem solving for a significant class of problems, so that its utility is not limited to stored 
problems; a coach for applying the problem-solving model (which can be turned off for free 
experimentation) that diagnoses errors and gives detailed suggestions at each step; a collection of 
instructive examples that is readily expandable by the teacher; a facility for entering arbitrary 
new problems; and an interactive guide to using the program, obviating the need for 
documentation. See:

Larkin, Jill. Robust Performance in Algebra: The Role of the Problem Representation. 
CDEC Reprint # 87-01. In C. Kieran and S. Wagner (Eds.) Proceedings of the Research 
Agenda Project on Algebra. Athens GA: National Council of Teachers of Mathematics (in 
press).

Schectic, Carol, David Trowbridge and Jill Larkin. Sketch. CDEC Report # 87-04. 

Trowbridge, David, Jill Larkin and Carol Schectic. Computer-Based Tutor on Graphing 
Conference (NECC'87). Eugene OR: International Council for Computers in Education, 
1987.

* Grapher & Calculator. Developed by Bruce Sherwood (CDEC and Physics) and Robert 
Schumacher (Physics) is a general graphing tool used by faculty in research as well as students 
that exploits the special functionalities of the Andrew windowed environment. The student may 
enter arbitrary parametric equations in standard algebraic form, including first-order ordinary 
differential equations. The program plots whatever variables the student chooses. Initial values 
and ranges may be changed at will. In calculator mode, the program computes values of 
user-defined functions. The program has an extensible, dynamic on-line help system that 
contains a catalog of problems and equations that can be clipped from the help window, pasted to the 
work window and executed for illustration.
Sierra is a generic multi-function calculator written in C by Kenneth Friend (CDEC).

* MAX is a program for matrix equations in linear algebra developed by Eugene Herman and Charles Jepsen at Grinnell College. Professor Herman spent 1987-88 in sabbatical at Carnegie Mellon as a visiting professor in the Mathematics department. CDEC/Sloan supported further work on MAX as well as a course on Computer Tools for Mathematical Problem Solving and a faculty seminar series on Computing in Mathematics Teaching by Professor Herman, which served both to enhance MAX and propagate the integration of computing tools like MAX in the mathematics curriculum. Professor Herman won a Distinguished Software and Curriculum award in July, 1988 from the EDUCOM / NCRIPtal Higher Education Software Program.

* Calculus Graphics and tutorial programs (six) were developed by Professor Russ Walker (Mathematics) for graphically illustrating and experimenting with the basic concepts of introductory calculus (over 600 students per semester); for the IBM XT with Hercules board and the advanced-function Andrew workstation.

GRADER is a computer-managed grading system for tracking and grading performance in Professor Peter Andrews' Educational Theorem Proving System (ETPS), an intelligent tutor for proving theorems in first-order predicate calculus. The software is sufficiently generic to be adapted to other on-line courses and its manual is available through CDEC. CDEC has also supported the development of ETPS.


2.3 The Humanities & Social Sciences

Some of the most major and ambitious CDEC and Sloan projects are in H&SS. Where major separate reports on projects exist, the projects are listed and the reports referenced.

2.3.1 English

* The WARRANT Project: Learner Centered Environments for Critical Reading, Reasoning, and Writing under Professors Preston Covey (Philosophy), Cheryl Geisler, David Kaufer and Christine Neuwirth (English) developed two programs exploiting the advanced-function workstation and campus network: Notes and Comments. See:


Far-reaching impact

We began with a writing task that is common yet crucial: read a group of authors who are addressing a single issue and make an original contribution to the issue. We took paternalism as
our original domain. (We defined paternalism as the issue of when, if ever, is it justified for society or another person to limit the freedom of an individual for that individual's own good.) After studying this issue in depth, we tested the generalizability of our findings in other domains: wilderness, animal rights, etc.

The task we identified raises the chances for the project's far-reaching impact for two reasons (1) it is central and (2) it is often not mastered well. First, the task is central to many intellectual activities: researchers carry out this task when they read the literature, plan a study and write up the results; managers carry out this task when they make arguments of policy; students carry out this task when they are asked to write an original paper. Second, it is often not mastered well: teachers complain that most students do poorly and employers welcome the rare individual who can do the task well.

Learner-centered

We did not begin with tool development. Although it might be appropriate to begin with tool development in domains where a great deal is known about problems learners experience with the task, we knew very little detail about the writing task just outlined or problems learners had with it. Thus, we began with a study of how experienced and inexperienced writers perform the writing task. We collected 40-50 hours of data apiece from experts and novices actually carrying out the task of reading a group of authors, reasoning about what they were reading, and saying something new. We analyzed what the task requires and what problems learners were experiencing.

This procedure put the learner and his/her problems at the center of the tool development, and significantly influenced the design of the tools.

Study the interaction of computer tools and curriculum

When people talk about the effects of computers, they often neglect the profound impact computer tools have on the curriculum. An obvious case is word processing and its impact on the writing curriculum. If a writing course moves to the computer lab, the nature of the curriculum is likely to be affected: it is more likely that students will spend time in class writing and that the teacher will spend less time lecturing and more time consulting with students about their in progress writing.

Thus, we have also collected and analyzed data of a teacher's writing class in which the lectures, discussions and assignments addressed the same task, in order to gauge how best to integrate the tools and classroom work.

On the basis of the data collection and analysis we designed a curriculum that we and others believe makes a significant contribution to teaching student writers how to read critically, reason about what they are reading and write original essays that make a contribution to an issue. We have a contract with Harcourt Brace Jovanovich (HBJ) to produce a textbook based on the curriculum. The book, *The Architecture of Argument: Transforming Issues through Reading and Writing*, is scheduled to appear in 1988. The innovations in curriculum required to exploit computer tools are reflected in the textbook.

Computer tool development

In parallel with the data analysis and curriculum development, we have developed computer tools to aid writers engaged in the processes of reading and writing. We have two of the tools deployed
in experimental sections of reading and writing courses on campus: Notes, a decision support tool for taking notes while reading, and Comments, a collaborative problem-solving tool for helping teachers and students respond to each others' texts.

Several principles have guided the development of these tools:

- Much of writing is typically invisible. For example, much of what is involved in building a synthesis traditionally resides primarily as a mental representation. There has been tremendous interest in systems that use visual representation to aid in programming, debugging, and understanding computer programs (Boecker, Fischer & Nieper, 1986). Along similar lines, we have designed visual representations and procedures that will provide an external representation for the knowledge structures needed in writing, together with operators for building those structures.

- It is now well-established that the right representation can significantly influence the ease of problem-solving (Simon, 1981). Our task analysis suggests that writers would do better to rely on not one external representation of memory but a variety of representations, with each representation providing a better match to a particular sub-task.

- Backtracking can be expensive, but it is characteristic of the creative design required in ill-structured problems (Perkins, 1981). We are working on designs that make backtracking less expensive.

- Writing actual prose represents a bottom-up planning procedure. As in other complex tasks, we observe a combination of top-down and bottom-up planning. We are trying to build our tools to support the easy conversion of prose to representation and from representation to prose.

- Although we have learned a great deal about the task, we need to learn more. The tools we have built and are building are instrumented to help us learn more about the task.

Create coherent patterns of learning across courses

Both Notes and Comments are being used in experimental sections of the following University core curriculum courses:

76-100 Strategies for Writing, the University Core course offered by the English department to introduce students to strategies for writing; required of all but Advanced Placement freshmen; 2 sections.

76-101 Reading Texts, a Designated Writing course offered by the English department to introduce students to strategies for reading literary and non-literary texts; one of several offerings for Advanced Placement freshmen; 2 sections.

76-122 Critical Writing, a Designated Writing course offered by the English department to introduce students to critical reading, writing and thinking, the curriculum developed in the WARRANT project; one of several offerings for
Advanced Placement freshmen; 1 section. There will be multiple sections in Fall, 1987.

Because the Notes and Comments program are intended to be used in a 15-16 week semester in writing courses, both are designed to be useful with minimum instruction, with increasingly sophisticated uses available. The programs are also self-documenting, with Help options available at major decision points.

Other groups on campus recognized the potential of the software to extend across clusters of courses, and the University has contributed support from the University's Writing Across the Curriculum program. With that support, we have been able to enhance the tools and also produce a Teacher/Student writing tool resource pamphlet that describes the use of these tools in any course that requires writing.

Perhaps even more importantly, teachers and researchers who write have recognized the utility of the tools. Some of the design of later versions have been influenced by demands from researchers rather than students.

Description of program

Notes
The CECE Notes program allows writers to take electronic notes from either online, electronic sources, or offline, printed ones. The Notes program is specifically designed to alleviate the difficulties and inconveniences that writers experience when they try to acquire and organize information from different sources. The program is currently in use in four sections of writing courses at Carnegie-Mellon.

Taking Notes with the CECE Notes Program
Notes allows writers to take notes from either online, electronic sources or offline, printed ones. For each note, the program automatically keeps track of the note's source. All notes are listed in a Main List that the program creates. From the All Notes List, writers may view notes they have already taken. They may view up to 16 notes on the screen at one time.

If a writer is taking notes from an online source, the Notes program links the section in the source from which the note is taken to the note itself and places an icon after the section in the source to which the note refers. Writers can access the note by activating the icon in the original source as well as from the program's Main List. In addition, since the program links the note to the online source, writers can refer to the original source of their notes quickly and easily from any note.

Writers can create classes of notes and assign notes to a specific class or classes. The program also allows writers to create , in addition to its Main List, any number of "alternative lists." With alternative lists, writers can create alternative arrangements of notes, including hierarchical ones. Both these facilities help writers to organize their notes in preparation for a writing task.

Writers can search in their directory of notes for general or specific information. For example, they may search for notes taken from a specific source, from a specific author, or in a particular class. Writers may also search for the name of a note or for a particular word or string in the notes. In addition, they may search on the time the notes were created or last
Implementation
Notes is currently implemented on the Andrew system, developed at the Information Technology Center, a joint computing venture between Carnegie-Mellon and IBM.

* Writers Workbench. Christine Neuwirth also ported ATT's Writer's Workbench to the Andrew system and redesigned its diction algorithm, integrating Professor Jaime Carbonell's (Computer Science) rule-based DYPAR program to improve the comprehensiveness and discrimination capabilities of WB's diction analysis.

2.3.2 History

* The Great American History Machine (GAHM) was developed by Professor David Miller (History) and Stephan Greene (CDEC). GAHM is a graphical data-mapping environment with generic machinery ('The Universal History Machine') that makes it adaptable to data mapping and analysis in history, geography, sociology, geography and macro-economics.

GAHM won the Distinguished Software award in the 1987 EDUCOM / NCRIPtal Higher Education Software Award Program. See:


2.3.3 Philosophy

* ANALYTICS is a software package developed on the IBM PC by Preston Covey and Steven Bend (CDEC).

The PC ANALYTICS package gives students practice in skills basic to formal logic and the analysis of arguments. It consists of three programs: Symbol, Truth, and Argue. Symbol generates drill exercises in symbolizing English sentences, and Truth in evaluating their truth value. Argue checks proofs and furnishes an environment for reconstructing arguments given in natural language into valid deductive form.

The SYMBOL Program

*Symbol* provides randomly generated exercises in symbolizing English sentences in a formal language for sentential logic. There are two basic types of exercise. In the first, the student is given a schematic English sentence, e.g., "P if and only if either Q or R." The student must identify the overall form of this sentence -- whether it is a conjunction, a disjunction, a conditional, or a biconditional -- and then symbolize the schema. The program will recognize the correct answer as well as a variety of basic errors and provide appropriate feedback in each instance.
In the second type of exercise, the student is given actual English sentences and assignments of their atomic components to sentential letters. Thus the sentence, "The Republicans will win the election only if the economy stays strong and neither Cuomo nor Bradley enters the race" is given with the following assignments, and the student must symbolize that sentence using the following assignments. Let:

- \( P = \) "The Republicans will win the election."
- \( Q = \) "The economy stays strong."
- \( R = \) "Cuomo enters the race."
- \( S = \) "Bradley enters the race."

The problems are randomly generated based on a library of atomic sentences, which can be edited at the instructor's discretion. The user has options allowing control over which kinds of logical connectives are used in the problems and how complex the problems will be.

The TRUTH Program

*Truth* provides randomly generated exercises in the truth-functional analysis of schematic English sentences. Each exercise begins with the presentation of a schematic sentence and the truth value of its components, e.g., "Either \( P \) or \( Q \), if and only if \( R \), where \( P \) is true, \( Q \) is true, \( R \) is false." Based upon the valuations given, the student must then determine whether the entire schematic sentence is true or false. Before making a determination, the student can see how the sentence would be symbolized, review the truth tables for the connectives, or build a truth tree for the given problem. If the student makes an incorrect determination, she is led step by step through a truth tree analysis to the correct truth value.

As with the *Symbol* program, the user has options which allow control over which kinds of logical connectives are used in the problems and how complex the problems will be.

The ARGUE Program

*Argue* both checks proofs and provides an environment for reconstructing English arguments in valid deductive form. In either mode, the student can work through problems stored in files or create problems of his or her own.

When using the program as a proof checker, the student works problems which consist of a conclusion and premises from which to derive that conclusion. The student moves towards the conclusion one step at a time, using any of the rules in the deduction system. (The program supports a wide variety of deduction systems, whose rules are entered in a file which can be edited by the instructor.) The proof checker will catch any errors as they are made. If the student is working a problem from a stored problem set, he or she can view any stored hints that the instructor has provided. There is no limit to the number of hints which can be provided with a problem.
Argument reconstruction problems consist of an actual English argument and assignments for translating the argument into a language for first-order predicate logic. The student must translate the argument into the formal language, providing any missing premises. The instructor can use stored hints to provide guidance on formulating missing premises. The student then uses the proof checker to prove the reconstructed argument valid by deriving the conclusion. At any time in the program, the student can view English translations, generated by the program, of every line in the proof.

The working environment consists of two windows adjustable in size, one for the derivation and one for problem statements, translations, hints, rule reviews, and other dialogue with the program initiated from pop-up help menus.

Requires IBM PC or compatible, minimum of 256K RAM; one double-sided floppy drive; monochrome or color display. See:


Covey, Preston, "Logic and Liberal Learning: Some Salient Issues", in Formal Logic and the Liberal Arts, a special double issue of Teaching Philosophy 4 (3/5) July/October 1981.

Covey, Preston, "Formal Logic and Philosphic Analysis", in Formal Logic and the Liberal Arts, a special double issue of Teaching Philosophy 4 (3/5) July/October 1981.


* CSYM: A Symbolization Tutor for First-Order Predicate Logic developed by Leslie Burkholder and Christopher Walton (CDEC) for the Andrew workstation. See:


* CMU Proof Tutor: an intelligent tutor for constructing natural deduction proofs in first-order logic developed by Professors Jonathan Pressler, Wilfried Sieg, Preston Covey and Richard Scheines (CDEC) for the Andrew workstation. See:


* TETRAD et al., a suite of expert-system programs for teaching probability analysis, statistical and causal modeling, and inductive logic developed by Clark Glymour, Kevin Kelly and Richard Scheines (Philosophy). The suite of AI and inductive logic software is integral to the Logic & Computation Program, which offers a major and a minor as well as graduate degrees in the Department of Philosophy, an interdisciplinary program that bridges formal logic, the philosophy of science, mathematics, statistics, social, cognitive and computer science. TETRAD itself is a state-of-the-art expert system for generating and testing competing causal models on very large data bases in the social sciences. See:


* PD World: a simulation environment for n-person iterated prisoner-dilemma models of the evolution of cooperative behavior and collective rationality, developed by Leslie Burkholder and Christopher Walton (CDEC) in cT.

PD World is interdisciplinary and strategic because it addresses an analytical approach that is hard to teach in general educational settings but that is crucial to the understanding of important work in several disciplines. It puts serious research tools in the hands of students as only the computer can: the environment is extensible, allowing a student to either replicate or extend all extant research (such as is described in Robert Axelrod's seminal book, The Evolution of Cooperative Behavior). PD World is seminal because it invites and propagates use across a spectrum of disciplines: philosophy, political science, decision science, economics, social psychology, policy analysis, socio-biology. These features have made it integral to courses in the Humanities & Social Sciences College, the Graduate School for Industrial Administration, and the School for Public & Urban Affairs: the intellectual substance of the program is integral to major research and problems for social institutions and public affairs. See:


2.3.4 Psychology

* The Lisp Tutor was developed by Professor John Andersen et al. (Psychology), another paradigm of the integration of cognitive research and educational computing in an exemplary intelligent tutor for teaching Lisp programming. See:


* **PsyLabs.** Fifteen simulation labs for cognitive psychology were developed by Professor Brian McWhinney. Published by Brookes Cole and deployed in the H&SS core course in Cognitive Processes & Problem Solving, the programs, which exploit the Andrew network and workstations as well as the IBM PC include classical experiments: Water Jugs, General Problem Solver, Tower of Hanoi, Logical Reasoning, Cryparithmetic, Comparing Stimuli, Signal Detection, Parallel Search, Short Term Memory Retrieval, Levels of Processing, Encoding Specificity, Paired Associative Learning, Method of Loci, Chaining, Keyword Mnemonics.

* **Experimental Design Tutorials.** Two tutorials to teach the process of experimental design were developed by Professor Margaret Clark for the IBM PC.

* **An on-line test generator,** grading and course management system with optical scanner were developed by Peter Lucas (Psychology) to expedite test administration and grading in self-paced psychology courses.

### 2.3.5 Social & Decision Sciences

* **The S Statistical System** package from Bell Labs was ported, adapted to the Andrew system with enhancement of its graphics capabilities by faculty in the Statistics department.

* **The Networker** and **Explorer** interactive packages were developed by Professor Kathleen Carley (Social & Decision Sciences) on a mainframe and adapted to the Andrew system to introduce students to network communication analysis techniques.

* **DEMOS** is an expert system for quantitative decision (risk, subjective probability, etc.) analysis in the social and policy sciences. Max Henrion, the developer of DEMOS, Charles Weicha, Dan Stoops (Engineering & Public Policy) and Greg Fischer (Social & Decision Sciences) developed **Giraffe,** a graphical plotter, and **Demaps,** a graphical tool for understanding and structuring quantitative decision models as complements to DEMOS for educational use on the Andrew system. This work is especially seminal now, given the burgeoning interest and need for education in the area of risk communication. See:


* PD World: a simulation environment for n-person iterated prisoner-dilemma models of the evolution of cooperative behavior and collective rationality, developed by Leslie Burkholder and Christopher Walton (CDEC) in cT.

PD World is interdisciplinary and strategic because it addresses an analytical approach that is hard to teach in general educational settings but that is crucial to the understanding of important work in several disciplines. It puts serious research tools in the hands of students as only the computer can: the environment is extensible, allowing a student to either replicate or extend all extant research (such as is described in Robert Axelrod's seminal book, *The Evolution of Cooperative Behavior*). PD World is seminal because it invites and propogates use across a spectrum of disciplines: philosophy, political science, decision science, economics, social psychology, policy analysis, socio-biology. These features have made it integral to courses in the Humanities & Social Sciences College, the Graduate School for Industrial Administration, and the School for Public & Urban Affairs: the intellectual substance of the program is integral to major research and problems for social institutions and public affairs. See:


3. Multi-Media Environments

Multi-media environments are singled out for special emphasis and attention, even though they also represent model applications. Two features distinguish our multi-media projects:

One is what is distinctive about multi-media environments themselves: the combination of full color, motion video and sound along with text and high-resolution graphics in a hyper-media format. Computer-based multi-media environments combine the data-richness and motivational powers of film or television with the interactivity and navigational powers of the computer. This combination of resources and powers poses both special opportunities and especially complex design and delivery problems. Hence, the need to treat the evolution of multi-media applications as a special strategic category.

The other distinctive feature of our multi-media projects is the initial arena of application: ethics, social issues and the arts -- domains of human values inquiry, typically neglected in the advance of educational technology to date.

No strategic vision of the evolution of educational technology can afford to ignore the promise and problems posed by developing multi-media technology (currently, interactive videodisc and CD-ROM technology, although these particular technologies are not definitive of the rapidly evolving state-of-the-art).

No strategic vision of educational computing can afford to ignore potential applications in the largely neglected domains of the humanities and the arts, those 'soft' areas of open-ended value-laden inquiry that seem to resist practical and rigorous methodologies delivered even by
traditional means or media. Curricula and pedagogy in these domains have never been so evidently important or so arguably in need of improvement.

For these reasons, CDEC has made a strategic priority of tracking and developing the promise and problems of multi-media technology and has chosen the particularly problematic staging area of values education for doing so. In the fullness of time, we will expand our efforts into the areas of science and technology education that are already exploiting computer-based multi-media.

**New Media for Values Education**

A critical look at technology in higher education should reveal a number of things, among them: undue expectations, mismanaged expectations, unfulfilled needs and promise, and needs and promise ignored. I will focus on the last issue, multi-media technology, and the theme of importing 'real world' information into education in ways that traditional media and methods cannot -- information that is rich, ambiguous, messy, difficult to measure or manage, but crucial.

The projects below address the specific, ill defined but now widely touted need for more attention to ethics and values in higher education. I will provide a framework for understanding some parts of that vast and complex need. Ethics and values education are not areas noted for exploiting either computer technology or 'real world' information, but they are areas where learning and inquiry can be improved thereby.

Among the messiest of 'information' to convey is the experience of valuation itself. Two of the videodisc applications briefly described below under Project THEORIA have demonstration videotapes available: A Right to Die? and Art or Forgy?

The application of multi-media technology to ethics and values education will raise more questions than it answers; but that's as it should be, since a sufficient benefit of the technology is to prove "a major stimulus for eliciting work and thought about teaching methods and how human beings learn" (Derek Bok, "Looking into Education's High-Tech Future," EDUCOM Bulletin, Fall 1985 -- a version of his 1985 annual report to the Harvard Board of Overseers, reprinted from Harvard Magazine, May/June 1985).

**3.1 Some Observations & Provocations**

**3.1.1 Lest Questions of Evaluation Evade Judgments about Value.**

*Five decades of research suggest that that there are no learning benefits to be gained from employing different media in instruction, regardless of their obviously attractive features or advertised superiority . . . . The best current evidence is that media are mere vehicles that deliver instruction but do not influence student achievement any more than a truck that delivers our groceries causes changes in nutrition.* [Richard Clark]
Like 99.99% of what passes for competent let alone venturesome practice in education, the
development and deployment of new technologies in education must largely proceed on the basis of
good judgment rather than rigorous (let alone conclusive) evaluation. Demonstrable gains for the
great costs involved are reasonable to demand, but these come only after considerable risk and
investment. The forms of demonstration are many, but few will pass muster with the evaluation
research industry. Derek Bok's seminal paper of 1985 (cited above) on technology in higher
education is a good overview of the types of value judgment we cannot evade by insisting on the
rigorous evaluative studies which few can afford. Evaluation research is not going to obviate the
risks and challenges confronting the principal inventors of educational innovation: educators.
The evaluation researcher's view (as distinct from the innovative practitioner's view) is
represented by Richard Clark's seminal article "Reconsidering Research on Learning from Media"
(Review of Educational Research, 53, 1983). Clark's most notorious conclusion was delivered
by the truck metaphor cited above.

Suppose we accept Clark's metaphor, hardly apt for learner-centered models of educational
process. By analogy, we might then look for technological vehicles for delivering various goods
where no other vehicles in fact are available, where the questions are not whether it's the
delivery vehicle that influences 'nutrition' but whether the goods are delivered at all and to what
ends.

Our concern is presumably with media that afford learning experiences and information of sorts
not available by traditional means. How demonstrable are the gains from new information or
media remains to be seen. But if we defer to instrumentalist evaluation paradigms like Clark's, we
will never learn. Technological innovation in education is not just about means and
instrumentalities for pedestrian instructional goals in tractable subject matters, it is not about
quantifiable instructional efficacy alone; it is also about the ill-defined ends of education,
open-ended learning processes and rich, ambiguous 'real world' information that is not easily
measured or managed.

For example, visual media (eg., photos, film, video) are not only alternative vehicles when it
comes to delivering essentially visual information; they are the only vehicles. The question is
when and why one needs pictures rather than other forms of data. Instruction is one thing;
inducing experience another. Instruction in ethics and value theory is one thing; learning by
inquiry and self-directed discovery is another. The questions for educators are when and why
certain information -- say, direct or simulated experience -- is needed. As we look critically at
technology in higher education and its future, we need beware the instrumental mentalities of the
evaluation gurus . . . ends, not means alone, are the matter for hard value judgments and there is
no science for evaluating ends. This paper is about elusive ends and messy means.

3.1.2 Neglected Domains: Open-Ended inquiry in the Arts and Humanities.

With all its powers, the computer cannot contribute much to the learning of open-ended
subjects like moral philosophy, religion, historical interpretation, literary criticism,
or social theory -- fields of knowledge that cannot be reduced to formal rules and
procedures. [Derek Bok]
This statement from Derek Bok's paper cited above is certainly true in its second claim. And the second claim may seem good reason for believing the first, if one's model is the computer as expert system and automated tutor, the computer as teacher. But if one takes seriously the model of the computer as navigational aid, data bank, simulation and experiential learning environment -- as persuasively described and positively assessed by Bok in the self-same paper -- then the first claim hardly follows. In fact, there are powerful heuristic-driven tutors, simulation and data-analysis environments to support and stimulate learning in many open-ended processes of decision, normative judgment, historical interpretation, literary (or art) criticism and social theory. And I will argue for an actual need for same in moral philosophy. Religion may be another matter.

Value and efficacy remain to be demonstrated. But there's no reason in principle to neglect such areas in critical assessments and creative exploration of technology in higher education. The fact that the very idea seems anathema or folly in so many quarters points to a failure of imagination about both means and ends that wisdom about either technology or education simply cannot abide.

3.1.3 Why Is Surgery Taught in Theaters, but Ethics Only in Theory?

*Non scholae, sed vitae discimus*

[Not for the school, but for life do we learn -- Epistolae morales]

Science, engineering, law, medical and business education -- various forms of professional or profession-oriented education -- seem to recognize this piety and necessity, at least in their respective domains, more fully than those humanities like ethics, which, with equal piety, aim to educate for life.

Put another way, no one would think of *credentialing* scientists, engineers, surgeons or other artists without providing them some hands-on view or experience with the actual stuff and rude realities of 'real life' practice; but regarding moral affairs, the groves of academe are experientially barren, devoid of the stuff of moral experience, with meager data in minute quantity, problems faintly viewed at grand theoretical distances, and propositions analyzed to a practically impotent fare-thee-well.

The study of ethics is typically academic and speculative in the worst senses of those terms. In serious science education we expect students to handle apparatus and process data that is rich in both quantity and quality. In ethics we typically rely on intellectual apparatus and demand neither quantity nor quality data; nor are we accustomed to introducing 'hard' or 'raw' data into our studies, being more comfortable with those abstract commodities of detached academic discourse -- propositions, reasons, definitions, concepts and denatured case studies.

Two hallmarks of engagement with reality, two commodities that pervade life and moral experience but are rarely entertained in ethics classes are: sense perceptions and feelings. There are at least two sorts of reasons for this discrepancy between the insipid stuff purveyed in classrooms and the strong stuff encountered in life outside -- one good, one bad.

The good reasons are practical. For example: Classrooms accommodate discourse well, but not many varieties of experience; the processing of articulate thoughts, but not confounding feelings.
The bad reasons are theoretical. For example: Moral issues are not about 'objective' physical phenomena, are not decided by observation, let alone anything like 'scientific' experiment, so perceptual data are not material. Moral reasoning and judgment are possible and credible only so far as they are 'rational,' and this means that feelings or emotion must either be vigorously rejected or rigorously subordinated. The textbooks say that appeals to emotion are fallacious, so why have them intrude in the first place?

The theoretical canon that dominates practice in the teaching of ethics has a venerable history. Fortunately for human life, no one pays much heed to the theory except in classrooms. Unfortunately, abstract ethical theory still dominates in the classrooms.

The issue of the role of emotion in moral reasoning and decisionmaking will provide grist for many academic mills for millenia; it has received increased attention in academic philosophy, as one of the emergent issues of the '80's. Every now and again an inordinately sensible philosopher will rescue the profession's reputation by siding with common sense and practical reality and proposing "a model for the mutual interaction of thinking and feeling in ethical decisionmaking" (Sidney Callahan, "The Role of Emotion in Ethical Decisionmaking," Hastings Center Report June/July 1988).

Unfortunately, the exigencies of life will not await the deliverances of this or any other generation's braintrust. Fortunately, there are other ways to learn than by studying the deliverances of academic theory.

For example, we can learn something about the role of emotion in moral reasoning by observation, by looking at our own experience and practice more directly, given good reflectors or simulators. We may then be in a position to invent our own theories of moral reasoning, if it's theories we need, much as theories were meant to be invented, by touching base occasionally with the realities about which they are theories.

There are venerable vehicles like literature and history and newer media like film and television for making our experience and practice accessible. One of the oldest media is theater, a medium characterized as capable of inducing the paradoxical phenomenon of detached engagement, a staple topic in courses on Greek Tragedy and a powerful attitude for learning.

Unfortunately, the study of Greek Tragedy as well as literature and film is itself more detached than engaged, though the analogues are instructive. Fortunately, yet newer technology provides yet another opportunity for engagement as well as detachment: a visual medium that depicts and induces palpable experience, but a medium with which we can interact -- computer-based multi-media. It is only ironic that we are driven to yet greater lengths to come closer to that touchstone of learning, experience.

'Experiential learning' is a bit of rhetoric invented to draw attention to forms of learning rescued from imprisonment in the ivory tower. Like 'applied ethics,' the expression is as useful as it is redundant.

To end with the appeal to authority with which I began . . . .

A sure sign that experiential learning, the affective dimension of 'real world' decisionmaking and video media are all on 'the right track' is that business schools -- who know from 'real life' and whereby their clientele's bread is buttered -- have taken up the cause. Drexel University's
School of Business introduced a curriculum that emphasizes affective awareness; see Vivian Rosenberg, "Affective Awareness as a Critical Thinking Skill," *Teaching Thinking & Problem Solving*, January/February 1986. The Business section of *INSIGHT* magazine, May 30, 1988 ran an item "Examination of Ethics by VCR" on Stanford University's School of Business Administration use of video case studies in the study of ethical dilemmas, with the following telling testimonial:

> Video helps capture the emotional element involved in these [insider trading] issues and provides a richer, more realistic presentation . . . Until now, Stanford solely used written case studies to examine questions of ethics.

Video case studies are employed with good reason in business, business schools and business ethics training programs and, as Derek Bok illustrated in his 1985 paper, the development and use of interactive video simulation environments is burgeoning in law and medical schools. Must we in the liberal arts always take our lead from the professional schools?

### 3.2 What Is Multi-Media?

**Multi-Media** here refers to any interactive computer-based environment that dynamically links several media: text, graphics, animation, sound, full-color high-quality video (motion or still). Computer-based multi-media environments combine the data-richness and motivational powers of film or television with the interactivity and navigational powers of the computer.

**Hyper-media** and **Inter-media** are also terms applied to computer-based environments that provide orderly, structured but non-sequential navigational links among a variety of media and data. The links are *hyper* in two dimensions: *vertical*, as in a multi-media computer-based encyclopedia, where one selects progressively 'deeper,' more detailed text and visual illustrations under a topic; and *horizontal*, where one leaps 'across' to any of many related topics, as one would from a *See also*: list in an encyclopedia. Hyper-media may have programmed links, or allow the user to navigate among perspicuously organized topics and resources building her own links.

*Interactivity* between the user and the environment (with varying degrees of *user control*) and *dynamic connectivity* among several media (on several levels) are features that typically distinguish computer-based multi-media environments from (1) *passively* viewed *linear* video programs on the one hand and (2) interactive computer environments limited to *monochrome text and graphics* on the other hand.

*Interactive video* and *laser optical media* are other terms for essentially the same concept. Many adopt *multi-media* (shorthand for 'computer-based multi-media') as the most generic of these various terms: *laser optical* denotes a certain technology; *video* denotes only part of the spectrum of data; *hyper-media* may beg the question of how deeply layered, dynamic, or interactive the environment really is; and *Inter-Media* happens to denote Brown University's implementation of this generic notion.

Multi-media environments may be implemented on a wide variety of computer-driven video or laser-optical technologies and hardware configurations: videotape, videodisc, CD ROM (compact disc read-only-memory), CDI (compact disc interactive), DVI (digital video interactive); with a video player or a stack of players beside the computer or out on a network; with two (or more)
Advanced Computer Applications for the Liberal Arts

screens (a CRT and color video monitor), or one color monitor (with text and graphics overlaying the video, with split screen, with co-lateral displays in juxtaposed, overlapping or stacked 'windows' on the screen, etc.). Unfortunately, these technologies and their markets are in inordinate flux.

CDEC's prototype multi-media applications have been developed on IBM XT-based systems but are in the process of being re-implemented in our own portable programming language, cT, for Macintosh- and Unix workstation-based systems. CDEC is working with MIT's Project Athena to adapt our applications to run in moveable, sizable windows on advanced-function workstations.

3.3 Why Multi-Media?

The limitations of a monochrome environment displaying only text and graphics (even with animation) are obvious: the more textual material provided, the less opportunity there is for interactivity; there is a clear limitation on the type of data that can be effectively presented: subjects and problems conducive to quantitative, abstract or schematic treatment clearly benefit; more concrete subjects dependent on observational evidence or perceptual data are less conducive to effective computer-interactive treatment.

The blessings of audio-video media are equally obvious, if mixed, as the power of film and television testify. The 'downside' of films and television for educational purposes is that these media are at once passively restraining and often merely overwhelming: they offer no means to stop the action, process one's thoughts or reactions reflectively, control the course and content of the presentation accordingly, or have one's responses to the presentation directly challenged.

Computer-driven interactivity, in theory, obviates these disadvantages. Interactive multi-media promises to combine both the wide bandwidth (the ability to communicate on many levels with diverse audiences) and the motivational power of audio-video media with the potential for orderly, structured, reflective inquiry -- ultimately under the user's control, with access to all manner of data and tools -- provided by the computer.

The advantages of computer-based multi-media environments are easily conjured, if not as readily realized:

Imagine... three fantasy scenarios: You are reading a Shakespearean text in one 'window' on a large high-resolution color monitor. You are drafting an essay for (a) your Shakespeare course (b) your history of tragedy course or (c) your stage design course. You're (a) arguing a particular interpretation of scene ii, act iii of Henry V (b) contrasting ancient Greek and Shakespearean theories of tragedy or (c) trying to design novel staging for the aforementioned scene.

With a few clicks of your 'mouse' you open another window that gives you ready access to available articles on whichever of these topics, cross-indexed and organized in hyper-text format, so you can browse and study myriad secondary source material at will from your workstation.

With a few more clicks you: (a) open two other windows and call up Lawrence Olivier's version of scene ii, act iii from the 1948 film in one, and Roderick Panchoff's version of the same scene from the video of the recent 1993 off-broadway production; (b) open two other window's to
contrast the performed denouement of Richard III with that of Oedipus Rex in the authentic BBC reconstructions of the 'original' performances; or (c) open another window to play scene ii, act iii of Henry V with a transparent color graphics overlay program that allows you to schematize the staging and action, which you save to a file for later inclusion as animated schematics in your essay.

In addition to footnotes and automated references in the electronic multi-media version of the 'essay' you send to your instructor, you include 'snapshot' sequences from the performances you use to argue your points.

For relief while working on your final draft, you plug your headset into an audio channel for a little stereophonic Vivaldi off the network's extensive collection. Or take a real break and call up Casablanca in a window from the film library's Bogart archives, painting mustaches on Ingrid Bergman with the graphics transparency package . . . .

This future is easy to conjure, but as difficult to implement as it is (willy nilly) inevitable. We need to begin to guide that inevitability to good effect. Apart from the fun and games, and the intrinsic power of the media to force this future, there is one singular bottom-line argument for multi-media:

Most of the real world and real-world problems for which we aim to equip our students are not well captured in books, lectures, class discussion, or even current interactive computer environments. The latter, for example, cannot readily simulate the practical realities or stimulate the human sensibilities that motivate and confound political or ethical dilemmas.

Typical students lack one important commodity for learning: life experience – or enough of it. Typical academic settings lack adequate means to provide this commodity. In academic terms, the lack is one of sufficient data and context -- necessities in any line of serious inquiry -- particularly in areas like ethics, the arts, or politics, where much of the essential data and context are perceptual, experiential, even emotional.

Multi-media environments are useful for the rich data, texture, and context they allow us to import into experientially barren groves of academic study -- allied with interactive computer technology for the easy control, flexible exploration, and disciplined reflection it can induce.

No strategic vision of the evolution of educational technology can afford to ignore the promise and problems posed by developing multi-media technology (typically, interactive videodisc and CD-ROM technology, although these particular technologies are not definitive of the rapidly evolving state-of-the-art).

No strategic vision of educational computing can afford to ignore potential applications in the largely neglected domains of the humanities and the arts, those 'soft' areas of open-ended value-laden inquiry that seem to resist practical and rigorous methodologies delivered even by traditional means or media. Curricula and pedagogy in these domains have never been so evidently important or so arguably in need of improvement.

### 3.4 CDEC Strategy & Priorities for Multi-Media

The world has become impatient for the 'revolution' the computer was to bring about for education.
Is the promise of multi-media just more premature techno-hype? How do we responsibly realize the promise of computer-based multi-media? Where do we begin, how do we proceed? What is our strategy?

Technology is often hyped as a solution to important problems, on the presumption that those problems are already well understood. Where the problems are educational, this presumption is too often defeasible.

This is ironic, in view of the hubris of the term technology, which, in its etymological roots, means the theory or logic (logos) of our know-how (techne').

For example, 'intelligent' computer-based tutors are often built on the assumption that humans understand the process of intelligent tutoring themselves; in most important areas, or in all but microscopic domains, we do not understand what it is exactly that humans do when they tutor (or learn) successfully. The overweening presumption that we understand the problems we address with technology is a source of gratuitous disappointment and misguided effort.

Strategies answer to problems. We begin by trying to identify the problems we want to understand and address. We do not presume that we yet understand these problems. We do assume that we can come better to understand and solve them through the adaptation of computer technology.

That is, we adapt technology in order better to understand (as well as to solve) an array of important problems for education and society.

Our first priority in COEC is to focus on the arts & humanities, the development of exploratory, experiential, data-rich, hands-on simulation environments for subject matters, skills, or problem domains that require experiential information that is rich in its appeal to our human sensibilities and, therefore:

1. pretty much neglected in the march of technology;
2. especially difficult (to design for) because the problems, methods, and operations we want to understand or teach are essentially ill-defined and contestable;
3. in need of 'live' data & rich context to support serious, hands-on inquiry or pro-active learning . . . and therefore
4. in need of good analogues, good complements to museum tours, field trips, hands-on lab, studio, clinical or 'real life' experience;
5. of wide social, cultural and public interest; and
6. exportable to public libraries, museums, schools, hospitals, and other settings outside the university accessible to the wider public.

Integration of the development of new educational technology and media with the agendas of both 'real world' institutions and the curriculum and intellectual life of the university is also a strategic priority for CDEC. Mechanisms for outreach, exchange and impact are two-fold:
As Vice Provost for University Studies, concerned with improving general education and core curricula across the university, I have recently emphasized a program called Values in the Arts, Sciences, Society & Technology (VAST), which initiates new interdisciplinary ethics and values curricula which test and use CDEC's new media.

As Director of Carnegie Mellon's new (effective June, 1988) Center for the Advancement of Applied Ethics (CAAE), I am developing a two-way channel between business, government and professional organizations in the 'real world' and curriculum development in the groves of academe. The CAAE offers external education programs to corporations, government and professional agencies and collaboratively develops new curricula with the university's professional schools and University Studies. Case materials developed collaboratively in our external education programs feed into university curriculum development, and vice versa. This two-way exchange with the real world will provide materials and resources for the development of multi-media environments well adapted to education for life in that world.
3.5 Project THEORIA: New Media for Values Education

The priority area of CDEC's multi-media work is a series of projects under the generic rubric Project THEORIA, whose agenda is reflected in its acronym: Testing Hypotheses in Ethics/Esthetics: Exploring the roles of Observation, Rationality, Imagination, & Affect. The goal of Project THEORIA is to design compelling, interactive simulation environments for testing hypotheses and 'theories' of the arts and morals -- among the most difficult and disputed of human value domains. (See projects listed below for examples on art forgery, the right to die, abortion.)

Theoria (Greek for theory) is also an allusion, to the concept of theory rooted in concrete observation, to the etymological roots of both theory and theater in the ancient Greek verb theorēin: to see, to view, to behold. Through exploitation of multi-media technology, we aim to provide a theater for ethical and esthetic theory, to bring the theory to ground in realistic settings that are rich in the complex data that any competent theory must first behold in order to explain.

In the 'Golden Age,' in the beginnings of the Western philosophical tradition in Greece, the vehicle for ethical theory was the theater: a spectacle, with universal elements of 'the human condition' reflected by chorus and convention in the concrete, compelling drama of Greek tragedy and comedy.

Theory in the arts and morals most naturally begins in what we see, imagine, or feel. Skills of moral reflection or reasoning, like the practical skills of the surgeon or the theoretic skills of the scientist, require an operating theater or laboratory for practice. We need good analogues of that theater or lab for the arts and morals: an experiential crucible for learning by seeing and doing.

Four projects are presently in various stages of concept, design or development for ultimate delivery on advanced workstations as well as microcomputers:

- A Right to Die? The Case of Dax Cowart (a videodisc, first in a series)
- Art or Forgery? The Case of Han Van Meegeren (a videodisc)
- Birth or Abortion? The Human Face of a Dilemma (mixed media, with videodiscs)
- Values Boggled: Ethics, Art & Money in the Work of J. S. G. Boggs (in concept)

These projects provide a mere sampler and only begin to illustrate the type and range of applications possible in the vast arena of human values inquiry.
3.5.1 **A Right to Die? The Case of Dax Cowart**

This prototype videodisc won a *Certificate of Merit* in the 1988 University of Nebraska Videodisc Award competition for its contribution to the state of the art.

**Content:** The videodisc presents the famous case of Dax Cowart -- a victim of severe burns, blindness and crippling injuries who persists under treatment to insist that he be allowed to die. Through interviews with Dax and other principals in the case (his doctors, lawyer, mother etc.), the user investigates basic ethical issues regarding quality of life, autonomy and competence, the obligations of medical professionals, etc. Throughout, the user must continually address the central dilemma: whether Dax should be granted his request to die - as well as the reasons why / why not.

**Features:** The videodisc will support eight or more hours of interactive exploration of the issues and case material, in two basic modes: 1. Access to video archives in which video segments are organized by both major issues and principals. 2. Socratically guided inquiry by which the user is led eventually to consider all the facts, issues and viewpoints. The program branches and questions the user in order to challenge her judgment and responses with contrary views and visuals. A NoteCard facility records the user's responses to questions or notes, organized under the relevant issue for output. The program uses these responses to direct the user to apt or challenging branches of inquiry and to query the consistency of her evolving views and judgments. When a final position is taken on whether to let Dax die, surprising consequences follow for either choice.

**Need:** Dax Cowart's request to die poses the kind of hard choice and hard case that makes or breaks our theories about what is right, best, or decent to do. Hard cases in ethics are born of rude realities, perplexing feelings & conflicted viewpoints. But those rude realities rarely invade the groves of academe & studied reflection is rarely afforded amidst the pressures of practical life. Our prototype videodisc aims to help bridge the gap between arrid theory & harried practice, to stimulate & simulate crucial conditions of critical moral reasoning in ways that other media cannot.

Critical moral reasoning requires, *inter alia*: empathy, the vivid representation of the interests of others; practiced confrontation with hard facts, unforeseen consequences & strong feelings; an appeal, at once, to our senses, sensibilities & minds; with opportunity for challenge & reflection.

Ethical theory must be brought to grips with issues in 'live,' affecting contexts, rich in the complex, perplexing data that any theory must first behold in order to explain. Theory or wisdom in morals begins most naturally in 'real' experience, in what we see, imagine or feel. *Skill* in ethical analysis or moral judgment requires the equivalent of a laboratory, studio or theater -- like the scientist's, artist's or surgeon's -- for 'safe,' hands-on, experientially rich *practice*.

For all these reasons, the study of ethics needs interactive video. And the world of interactive video technology needs a 'proof of concept' project, an *experimentum crucis* to show that it can serve the cause of education in ethics and address salient, pressing social issues.

**Audience:** A crucial test is by our colleagues in law, medicine and health services; our disc is intended to be useful in settings of professional education and practice, where our prototype is presently being reviewed and site tested. Prime targets are postsecondary teachers & students, with whom it was used this semester. And because of the wide 'bandwidth' of the medium, its
power to communicate on several levels with diverse audiences, we also see the disc as a resource for public schools & libraries.

3.5.2  **Art or Forgery? The Case of Han Van Meegeren**

This videodisc does for aesthetics and art history studies what the other projects attempt to do for education in applied ethics: to put interactive computer tools and compelling, realistic data for intensive hands-on inquiry into the hands of users (be they undergraduates in aesthetics or art history courses, high school art teachers or students, or members of the museum-going public).

The videodisc raises compelling, generic issues in aesthetics, art history or 'art appreciation' in the context of a dramatic 'real life' art-world scandal that occurred at the end of WWII. It is designed for university, school, or public deployment (e.g., in libraries and museums), by teachers in class settings (for interactive presentation and discussion) or by individuals for self-study.

The videodisc program is set up as a 'detective story,' putting the user in the role of an investigative reporter who must sift through and weigh the historical, scholarly, visual and forensic evidence to determine whether a given painting (sold to Nazi Hermann Goering during WWII by a third-rate Dutch painter, Han Van Meegeren) is in fact an original Vermeer (as claimed by the art experts) or a forgery (as claimed by Van Meegeren, in order to escape a life sentence as a Nazi collaborator). Aesthetic issues are also raised about the status of a good forgery as 'art' and what makes any work a work of art. The program is designed to exercise the user in the critical, observational and analytic skills required of any informed observer of visual art. The videodisc makes resources from museums and libraries around the world readily accessible.

The goals of the videodisc are to provide (1) easy access to a rich store of information and art work in one convenient place to provoke (2) close attention to details and features of art works and (3) analysis of the evidence pertaining to issues of attribution and aesthetic value to enhance the observational and analytic skills needed to appreciate visual art, to generate and test hypotheses under duress from contradictory data.

Issues of attribution are heuristic vehicles for basic lessons in the interdependence of fact and value judgments, the nature of human value judgment and evidentiary standards, and the weighing and balancing of protean 'evidence' that can at once compel and mislead in the conflict of alternative explanations.

The videodisc has been deployed with 90 students in our Aesthetics course and will be showcased at several major conferences in 1988-89: in art education, philosophy, the American Association of Museums, the College Art Association and LaserActive '88, an annual national conference on new developments in laser optical media. For detail information see:

Covey, Preston K., Lisa Leizman and Kate Maloy. **Art or Forgery? The Case of Han Van Meegeren - A Videodisc.** CDEC Report # 88-07.

Covey, Preston K., Lisa Leizman and Nicholas Spies. **Art or Forgery? The Case of Han Van Meegeren. A videotape demonstrating the prototype videodisc.** CDEC videotape # 88-12.
3.5.3 *Birth or Abortion? The Human Face of a Dilemma*

This project has just begun in the prototype design, 'proof of concept' phase with the production of sample video case materials in linear and interactive formats, for purposes of seeking funding. The project will develop original educational material in several media for public, secondary and postsecondary audiences to address a moral and social problem of the first order: the conflict and controversies surrounding the issue of abortion. The project and its case materials are derived from well advanced interview research and a book in preparation by Kate Maloy and Margaret Patterson, *Birth or Abortion? Private Struggles in a Political World*.

The approach is to develop a painstakingly balanced set of case materials and multi-media resources for exploring the complex dimensions, moral perplexities and human realities of the problems of human choice. There is no attempt to argue any one moral, ethical, political or sociological viewpoint. This intensive case study approach is modelled on that of J. Anthony Lucas’ well received and highly acclaimed study of the school desegregation controversy in Boston, *Common Ground: A Turbulent Decade in the Lives of Three American Families* -- viewing the realities and complexities of the dilemmas through the eyes of participants and professionals on all sides of the larger social problem.

The educational goals of the project are basic lessons in moral inquiry and imagination, in the spirit of the following observations on cognate projects, but with the added impact afforded by the visual, interactive medium -- lessons that nobody knows yet how to measure in any medium:

*The three families at the center of my story were not selected as statistical averages or norms. On the contrary, I was drawn to them by a special intensity, an engagement with life, which made them stand out from their social contexts. At first, I thought I read clear moral imperatives in the geometry of their intersecting lives, but the more time I spent with them, the harder it became to assign easy labels of guilt or virtue. The realities of urban America, when seen through the lives of actual city dwellers, proved far more complicated than I imagined.* [J. Anthony Lucas]

*I try to ... bring the reader up close, so close that his empathy puts him in the shoes of the characters. You hope when he closes the book his own character is influenced.* [William Carlos Williams]

Through a mix of media, the project aspires to bring its audience -- traditional and non-traditional learners alike -- 'up close' to human realities, moral perplexities and conflicted sensibilities that often confound our best efforts to chart and lead decent human lives. All the project's components, the interactive media in particular, aim to induce one incontestably crucial condition of competent moral reflection: the vivid representation of the interests of others, the appreciation of moral, emotional and practical straits that -- but for fickle fate or feeble imagination -- afflict us all, our common human ground for negotiating conflict in community.

If the project finds funding, its products will ultimately include: the book of case studies (of professionals as well as women and men who have struggled with different problems, decisions and their consequences), a public broadcast television production, a set of educational videotapes, videodisc archives of case and background material, interactive programs highlighting ethical and gender issues, and accompanying study guides. For details, see Preston K. Covey, *Birth or Abortion? The Human Face of a Dilemma - A Multi-Media Project*. CDEC Report # 88-01.
3.5.4 **Values Boggled:**

*Ethics, Art & Money in the Work of J.S.G. Boggs*

This interactive videodisc concept was provoked by the work and trial of the conceptual artist J.S.G. Boggs, recently reported in a fascinating serial article by Lawrence Weschler in the *New Yorker* ("Values I & II," January 18 and 25, 1988 -- excerpted from Kapinsky's *Karma & Boggs's Bills*, San Francisco: North Point Press, 1988).

Boggs does very exacting life-size replicas of paper currency (one side only, with documentation on the other side, so that there is no intention of counterfeiture), which he then tries to "spend," with no attempt to defraud or pass off his product as actual currency. Therein hang many tales.

Boggs, whose artwork consists in the transactional inquiries that result from his attempts to "spend" his artifacts, was tried (unsuccessfully) by the Bank of England for his precise reproductions of pound notes. Therein hangs another tale.

Boggs' work and story raise countless intriguing questions about money, "art" and "value," around which a multi-media, multi-dimensional curriculum in values inquiry could be built.

In Boggs' view, his art form does not consist of the artful replication of currency but rather in the whole (series of) transactions that his attempt to "spend" his product generates: his project is really a transactional inquiry into fundamental questions of art, value and ethics. The artifactual elements of his art works include all the paraphenalia (from change and receipts to his actual "purchases") that result from and evidence his transactions; these are what Boggs collectors collect. But his practice stands as a paradigm of conceptual art and challenges the conceptual limits between art and life. Boggs also exemplifies the crucial comedic dimension of serious values inquiry.

This case raises very basic, compelling issues in ethics, aesthetics, value theory, economics, social history and law: deep questions about the nature of value, the nature and value of art and money, the cultural norms that govern their relationships, and the ethos of the artist in society. The visual interests and transactional drama of this case lends itself naturally to interactive video treatment.

The case provides a nexus of issues for interdisciplinary studies. There exists a rich social history of artists' reproductions of currency, for various and nefarious purposes, a thematic thread for social historical focus. There are interesting philosophic questions (dramatized in his trial and subsequent hassles) for the law in Boggs' case. While his case is not strictly one of either fraud or forgery, it would make a natural progression from such cases into more fundamental issues regarding what is "art" or "value" (that begin dramatically when people either agree or refuse to accept his artwork in lieu of money). One interesting subtext in the case is the history and nature of "money," monetary or fungible value, and the monetary system -- a natural segway into the philosophy of economics as well as the economics of the art world.

This project is just entering the design treatment and grant proposal phase. Boggs and Lawrence Weschler, a philosopher and professional writer for the *New Yorker*, have expressed interest in collaborating with CDEC on the eventual project, which will, in effect, become an extention of Boggs' artistic and educational work in a new medium.
### Sloan Grant Summary 1984 - 88

For Carnegie Mellon Fiscal Years (July 1 - June 30)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Beginning Balance</td>
<td>178,482</td>
<td>43,276</td>
<td>37,637</td>
<td>107,248</td>
</tr>
<tr>
<td>Income</td>
<td>75,000</td>
<td>100,000</td>
<td>100,000</td>
<td></td>
</tr>
<tr>
<td>Transfers</td>
<td></td>
<td></td>
<td>30,592</td>
<td></td>
</tr>
<tr>
<td>Total Expenditures</td>
<td>210,206</td>
<td>136,231</td>
<td>30,389</td>
<td>107,248</td>
</tr>
</tbody>
</table>

**PERSONNEL**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Administration</td>
<td>12,499</td>
<td>18,169</td>
<td>1,075</td>
<td></td>
</tr>
<tr>
<td>Computer Skills Workshop</td>
<td>15,830</td>
<td>12,127</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation</td>
<td>7,500</td>
<td>2,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity Tools (cT &amp; GLO)</td>
<td>34,140</td>
<td></td>
<td>27,350</td>
<td></td>
</tr>
<tr>
<td>Fine Arts</td>
<td>2,500</td>
<td>2,500</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sciences &amp; Mathematics</td>
<td>34,180</td>
<td>21,989</td>
<td>2,000</td>
<td>11,131</td>
</tr>
</tbody>
</table>

**Humanities & Social Sciences**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Psychology Suite</td>
<td>58,495</td>
<td>8,052</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WARRANT - Writing Environments</td>
<td>10,500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Great American History Machine</td>
<td>3,240</td>
<td>6,655</td>
<td>10,010</td>
<td></td>
</tr>
<tr>
<td>Social Science Suite</td>
<td>9,600</td>
<td>7,416</td>
<td>4,629</td>
<td>13,372</td>
</tr>
<tr>
<td>Logic &amp; AI Suite</td>
<td>2,616</td>
<td>33,099</td>
<td>3,579</td>
<td>20,606</td>
</tr>
<tr>
<td>Multi-Media for Values Education</td>
<td>4,616</td>
<td></td>
<td>15,135</td>
<td></td>
</tr>
</tbody>
</table>

**BENEFITS**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>21,395</td>
<td>9,138</td>
<td>3,913</td>
<td>19,629</td>
</tr>
</tbody>
</table>

**OPERATING & EQUIPMENT**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8,211</td>
<td>4,586</td>
<td>566</td>
<td>25</td>
</tr>
</tbody>
</table>
**Sloan Grant Detail 1984 - 88**

For CMU Fiscal Years (July 1 - June 30)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PERSONNEL</strong></td>
<td>180,601</td>
<td>122,507</td>
<td>25,910</td>
<td>87,594</td>
</tr>
<tr>
<td>Administration</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crecine PI 10%</td>
<td>8,884</td>
<td>10,458</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Secretary 20%</td>
<td>2,529</td>
<td>1,503</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Student/Research Asst's</td>
<td>1,086</td>
<td>6,208</td>
<td>1,075</td>
<td></td>
</tr>
<tr>
<td>Computer Skills Workshop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stuckey, CSW Director 50%</td>
<td>15,830</td>
<td>12,127</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Documentation (CDC)</td>
<td>7,500</td>
<td>2,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Productivity Tools (cT &amp; GLO)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sherwood PI 25%</td>
<td>15,000</td>
<td></td>
<td>17,375</td>
<td></td>
</tr>
<tr>
<td>Programmers</td>
<td>19,140</td>
<td></td>
<td>9,975</td>
<td></td>
</tr>
<tr>
<td>Fine Arts</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Marilyn Thomas VOICE</td>
<td>2,500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gerhardt Schmitt PRINCE</td>
<td>2,500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sciences &amp; Mathematics</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trowbridge G&amp;T et al.</td>
<td>13,750</td>
<td>1,354</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Richman</td>
<td>4,480</td>
<td>685</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VanderVen</td>
<td>3,200</td>
<td>250</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Karol</td>
<td>5,000</td>
<td>5,000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bothner-By</td>
<td>5,250</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Walker</td>
<td>2,500</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Schumacher</td>
<td></td>
<td>3,840</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kaufman/Sherwood</td>
<td>3,200</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kiester</td>
<td>3,840</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Henein</td>
<td>3,840</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patterson</td>
<td></td>
<td>2,000</td>
<td></td>
<td>9,131</td>
</tr>
<tr>
<td>Yee/Whitely Sketch</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Herman Math Tools</td>
<td></td>
<td></td>
<td></td>
<td>2,000</td>
</tr>
</tbody>
</table>
### Humanities & Social Sciences

#### Psychology Suite

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISP Tutor</td>
<td>27,238</td>
</tr>
<tr>
<td>Experimental Design Lab</td>
<td>19,771</td>
</tr>
<tr>
<td>Cognitive Psych Labs</td>
<td>11,486</td>
</tr>
<tr>
<td>On-line Test Generator</td>
<td>1,500</td>
</tr>
</tbody>
</table>

#### WARRANT - Writing Environments

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Great American History Machine</td>
<td>3,240</td>
</tr>
</tbody>
</table>

#### Social Science Suite

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>S Stat package on Andrew</td>
<td>2,000</td>
</tr>
<tr>
<td>Carley NETWORKER et al.</td>
<td>7,600</td>
</tr>
<tr>
<td>Henrion DEMOS</td>
<td>7,416</td>
</tr>
<tr>
<td>Burkholder/Walton PD World</td>
<td>4,629</td>
</tr>
</tbody>
</table>

#### Logic & AI Suite

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bend et al. ANALYTICS</td>
<td>1,365</td>
</tr>
<tr>
<td>Burkholder/Walton CSYM</td>
<td>1,251</td>
</tr>
<tr>
<td>Glymour AI &amp; Inductive Logic</td>
<td>12,140</td>
</tr>
<tr>
<td>Kelly VALID on-line logic course</td>
<td>2,000</td>
</tr>
<tr>
<td>Sieg DESIGN Volume &amp; Proof Tutor</td>
<td>8,167</td>
</tr>
<tr>
<td>Miller/Scheines/Walton Proof Tutor</td>
<td>2,865</td>
</tr>
</tbody>
</table>

#### Multi-Media for Values Education

<table>
<thead>
<tr>
<th>Software</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4,616</td>
</tr>
</tbody>
</table>

#### Benefits

<table>
<thead>
<tr>
<th></th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>BENEFITS</td>
<td>21,395</td>
</tr>
<tr>
<td></td>
<td>9,138</td>
</tr>
<tr>
<td></td>
<td>3,913</td>
</tr>
<tr>
<td></td>
<td>19,629</td>
</tr>
</tbody>
</table>

#### Operating & Equipment

<table>
<thead>
<tr>
<th>Operation</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Supplies, copying etc.</td>
<td>487</td>
</tr>
<tr>
<td>Data Processing, computing etc.</td>
<td>1,555</td>
</tr>
<tr>
<td>System Software</td>
<td>1,190</td>
</tr>
<tr>
<td>Hardware, peripherals</td>
<td>4,169</td>
</tr>
<tr>
<td>Travel, speakers etc.</td>
<td>810</td>
</tr>
</tbody>
</table>
CDEC Technical Report Series

Preston K. Covey, Director
Steven Bend & Carol Scheftic, Editors

Center for Design of Educational Computing
Carnegie Mellon University
Pittsburgh, PA 15213

August 21, 1988
1985


1986


86-02 Burkholder, L. (June, 1986). The Halting Problem.


86-04 Covey, P. (June, 1986). New Technology & Directions for Educational Computing in Philosophy. Presented at the Ohio State University Dept. of Philosophy, Columbus OH.


1987


87-04 Scheftic, C., Trowbridge, D., & Larkin, J. (May, 1987). Sketch. Final application submitted to the EDUCOM/NCRPTAL Higher Education Software Awards Program. (Sketch received a "Distinguished Software" award in October, 1987.)


87-06 Covey, P. and Sobel, A. (May, 1987). The Andrew/Kyoto Common Lisp Project.


87-17 Covey, P. (October, 1987). CDEC Multi-Media Prospectus.


1988


88-10 Burkholder, L. (June, 1988). Adding to a File in PROLOG.

88-11 Covey, P., Roberts, S. & Spies, N. (June, 1988). *A Right to Die? The Case of Dax Cowart*. A videotape demonstrating the videodisc submitted to the University of Nebraska 1988 Interactive Videodisc Awards program. 1/2" VHS and 3/4" Beta formats. (The videodisc was awarded a Certificate of Merit in July 1988.)


Covey, P. (Editor) (August, 1988). *The Role of Design in Liberal/Professional Education*.


Covey, P. (September, 1988). New Media & Values Education. To be presented at and published in the proceedings for the Alfred P. Sloan Foundation invitational conference *A Critical Examination of Technology in Higher Education: Importing Real World Information into Education*, Dartmouth College, September 30 - October 2, 1988.