Project IMPACT is a comprehensive advanced development project designed to produce an effective and economical computer-administered instruction (CAI) system for the Army. This report describes the concepts, approach, and implementation of the Project IMPACT text-handling subsystem. The computer-based facilities for preparing, storing, and retrieving the content of CAI courses of instruction are described. Computer software tools, called EDITOR and DIRECTOR, are described in terms of their use by course authors. EDITOR enables authors to prepare the content of instruction easily and rapidly, while DIRECTOR manages the instruction presented to students by the computer. (Author/JK)
Technical Report
71-21
Project IMPACT—
Computer-Administered Instruction:
Preparing and Managing the
Content of Instruction,
IMPACT Text-Handling Subsystem

The IMPACT Staff

HUMAN RESOURCES RESEARCH ORGANIZATION
300 North Washington Street • Alexandria, Virginia 22314

September 1971
Destroy this report when it is no longer needed.
Do not return it to the originator.
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4. Title and Subtitle</td>
<td>PROJECT IMPACT--COMPUTER-ADMINISTERED INSTRUCTION: Preparing and Managing the Content of Instruction, IMPACT Text-Handling Subsystem</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7. Author(s)</td>
<td>The IMPACT Staff</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9. Performing Organization Name and Address</td>
<td>Human Resources Research Organization (HumRRO) 300 North Washington Street Alexandria, Virginia 22314</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12. Sponsoring Organization Name and Address</td>
<td>Office, Chief of Research and Development Department of the Army Washington, D.C. 20310</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15. Supplementary Notes</td>
<td>HumRRO Division No. 1 (System Operations), Alexandria, Va., Work Unit IMPACT</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16. Abstracts</td>
<td>Project IMPACT is a comprehensive advanced development project designed to produce an effective and economical computer-administered instruction (CAI) system for the Army. This report describes the concepts, approach, and implementation of the Project IMPACT text-handling subsystem. The computer-based facilities for preparing, storing, and retrieving the content of CAI courses of instruction are described, as are CAI courses. Computer software tools are described in terms of their use by course authors.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 17. Key Words and Document Analysis. 17a. Descriptors | *Man machine systems  
*Educational technology  
*Computer software  
Individual instruction  
Training technology  
Cathode ray tubes |
| 17b. Identifiers/Open-Ended Terms | Computer-assisted instruction  
CAI author languages |
| 17c. COSATI Field/Group | 0902 Electronics and electrical engineering Computers |
| 18. Availability Statement | 3 |
| 19. Security Class (This Report) | UNCLASSIFIED |
| 20. Security Class (This Page) | UNCLASSIFIED |
| 21. No. of Pages | 47 |
| 22. Price | USCOMM-DC 4038-P71 |
The Human Resources Research Organization (HumRRO) is a nonprofit corporation established in 1969 to conduct research in the field of training and education. It is a continuation of The George Washington University Human Resources Research Office. HumRRO's general purpose is to improve human performance, particularly in organizational settings, through behavioral and social science research, development, and consultation. HumRRO's mission in work performed under contract with the Department of the Army is to conduct research in the fields of training, motivation, and leadership.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.
FOREWORD

This report describes the concepts, approach, and implementation of the Project IMPACT text-handling subsystem, which assists in preparing and managing the content of computer-administered instruction (CAI) courses. Work Unit IMPACT, Prototypes of Computerized Training for Army Personnel, is an advanced development project undertaken by the Human Resources Research Organization to provide the Department of the Army with a CAI system. The National Science Foundation is also sponsoring HumRRO research on Instructional Decision Models (IDMs), with additional support provided by the James McKeen Cattell Fund.

This document is a second report on the progress of the hardware/software subsystem (specifically the text-handling facilities) toward implementing the initial IDM within the HumRRO CAI environment. The first-generation subsystem was described in Project IMPACT: Computer-Administered Instruction, Description of the Hardware/Software Subsystem, HumRRO Technical Report 70-22.

The research is being conducted at HumRRO Division No. 1 (System Operations) Alexandria, Virginia, where Dr. J. Daniel Lyons is Director. Dr. Robert J. Seidel is the Program Director.

The IMPACT text-handling subsystem was developed by the Project IMPACT staff, principally Mr. Jean Garneau, Mrs. Doris Shufield, Mr. Leslie W. Willis, and Mr. John Stelzer. This report was prepared by Mrs. Beverly Hunter, Mr. Martin Rubin, and the Project IMPACT staff.


Identification of products is for research documentation purposes only, and does not constitute an official endorsement by HumRRO, the Department of the Army, the National Science Foundation, or the James McKeen Cattell Fund.

HumRRO research for the Department of the Army is conducted under Contract DAHC 19-70-C-0012. Computer-Administered Instruction research is conducted under Army Project 2Q063101D734. The IDM research being conducted under National Science Foundation sponsorship is funded under Grant GJ-774, Research on Instructional Decision Models, with additional funds from the James McKeen Cattell Fund.

Meredith P. Crawford
President
Human Resources Research Organization
BACKGROUND

The objective of Project IMPACT is to evolve a series of prototype systems of Computer-Administered Instruction (CAI) in order to produce a total CAI system that is effective, efficient, and cost-effective for operational use in Army training. The total prototype system includes four main components:

1. Hardware—The computer, student stations, and related equipment.
2. Software—The computer programming systems that control operation of the hardware.
3. Courses of Instruction—The actual content and logic of courses administered by the computer.
4. Instructional Decision Models (IDMs)—The rules and strategies by which specific course content is provided to an individual student.

The way in which the content of instruction is prepared, stored in the computer, and managed by the computer while students are taking a course has a great influence on the efficiency and effectiveness of the total CAI system. All the hardware and software components of the system which together provide facilities for preparing, storing, and managing the content of instruction are called the “Text-Handling Subsystem.”

OBJECTIVES OF THE TEXT-HANDLING SUBSYSTEM

The principal objectives of the text-handling subsystem are twofold:

1. To provide a rapid, easily used means for course authors to prepare and maintain the content of courses of instruction. The costs of course development should be minimized by allowing content specialists to prepare instructional materials without regard to technical hardware and software considerations. Members of the course development team should be able to enter in the computer, review, and revise materials without going through intermediaries such as computer programming languages, a special programming staff, or computer center procedures.
2. To provide a means for storing and retrieving course content that will maximize flexibility of course design and instructional strategies. The evolution of effective strategies for teaching via computer requires that authors and researchers have the greatest possible flexibility for experimenting with various strategies and course design.

Another objective is to allow for flexibility in the kinds of computer programming languages used to implement course strategies and logic. The text-handling subsystem should allow for experimentation with various kinds of computer programming languages that may be suited to implementing particular types of instructional strategies and the IDM.

APPROACH

The IMPACT approach to achieving the foregoing objectives is to provide a text-handling subsystem in which the content of a course—the text—is stored in files that are physically separate from the computer programs that control course strategies or course logic (i.e., an IDM and its interpretation in a computer programming language). Course

authors can enter the text into the computer files directly through the CRT, without writing any computer programs. Similarly, the text can be rapidly and easily retrieved (viewed on the CRT screen), reviewed, modified, and so forth, directly through the CRT terminal.

When students are taking a course, the text-handling subsystem retrieves the appropriate text from the files, arranges it in appropriate sequence for that student, and transmits it to the student's CRT. The text-handling subsystem functions under commands from the computer programs containing the logic, or strategy, of a course (the IDM).

The text-handling subsystem is designed so that it can be implemented in a variety of computer systems, that is, to be compatible with a variety of programming languages, terminal hardware devices, and IDMs. In IMPACT's first-generation CAI system, the text-handling subsystem is implemented on an IBM 360 model 40 computer, using the Sanders 720 CRT terminal and the CAI programming language Coursewriter III.¹

FUNCTIONS

The IMPACT text-handling subsystem performs or supports the following functions:

1. **Text preparation and maintenance.** The content of courses is created, improved, and maintained with the assistance of EDITOR. EDITOR (Entry on Disk of Instruction Text for Online Retrieval) is a simple, powerful set of commands that enables members of the course development team to create and modify text. FACS (File Activity Control System) is another tool that assists in preparing and maintaining courses. FACS is a set of computer programs that produces various administrative and technical reports regarding the course materials stored on the text files.

2. **Text storage.** The text is stored in small elements (usually filling less than one CRT screen) on text files at the central computer. The text may be grouped and indexed on these files in any way that is appropriate for a particular course or IDM (e.g., all practical exercises might be stored on one file). The number of such files established is logically unlimited.

3. **Text retrieval.** Elements of text are retrieved from the files and arranged in various ways to produce instructional units. DIRECTOR is a software supervisor that accepts commands from IDM as to what text is needed for each individual student at given points in the course. DIRECTOR retrieves the text from the files and transmits it to the student's CRT. Commands to select and display moving or single frame film presentations are also transmitted by DIRECTOR.

STATUS

The text-handling subsystem as described in this report is operational at Project IMPACT facilities. It is part of a first-generation CAI system, not intended for widespread, operational implementation at Army schools. The efficiency and effectiveness of the approach to text handling is being tested in this first-generation system. If the approach proves to be viable, the subsystem implementation will be extended in the next generation to accommodate a wider variety of hardware devices, programming language interfaces, and IDM.

¹ Identification of products is for research documentation purposes only, and does not constitute an official endorsement by HumRRO or the research sponsors.
# CONTENTS

<table>
<thead>
<tr>
<th>Part</th>
<th>INTRODUCTION</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Background</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>Text-Handling Considerations in CAI</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Objectives of the IMPACT Text-Handling Subsystem</td>
<td>5</td>
</tr>
</tbody>
</table>

| 2    | OVERVIEW OF IMPACT TEXT-HANDLING SUBSYSTEM | 7 |
|      | General Characteristics | 7 |
|      | Course Development | 7 |
|      | Course Administration | 8 |

| 3    | PREPARING INSTRUCTIONAL TEXT USING EDITOR | 11 |
|      | The Six EDITOR Commands | 12 |
|      | CREATE (CRE) | 13 |
|      | MODIFY (MOD) | 13 |
|      | DELETE (DEL) | 13 |
|      | COPY (COP) | 14 |
|      | DISPLAY (DIS) | 14 |
|      | ROLL ON (ROL) | 14 |
|      | Roll Ons and Blocking | 15 |
|      | Blocking | 15 |
|      | The Roll On Command | 17 |
|      | Multiple Commands | 17 |
|      | Preparing a Series of Displays for Online Presentation | 18 |
|      | The Significance of EDITOR | 20 |

| 4    | THE TEXT FILES | 21 |
|      | Multiple Files | 21 |
|      | Present IMPACT Text Files | 21 |
|      | Relation of Text Files to Courses | 22 |
|      | Information in the Files | 22 |
|      | Retrieval Identifiers | 22 |
|      | File Integrity and Security | 22 |

| 5    | FACS (FILE ACTIVITY CONTROL SYSTEM) | 23 |
|      | FACS Management Uses | 23 |
|      | FACS Text Development Uses | 23 |
|      | Text Documentation and Worksheets—The Image Reports | 24 |
|      | A Series of Text Elements—The Dump Text Reports | 24 |
|      | An Analytical Tool—The String Reports | 27 |
|      | Administrative Reports | 27 |
Part
6  HOW THE COMPUTER HANDLES TEXT DURING COURSE ADMINISTRATION ........................................ 28
   DIRECTOR and Course Programs ................................................................. 28
   DIRECTOR and CRT Hardware ................................................................. 29
   DIRECTOR and Course Authors ............................................................... 29
      Dynamic Text Creation ........................................................................... 30
      Conversational Interaction ...................................................................... 31
      Student Inquiry ....................................................................................... 31
      Generalized Course Logic ...................................................................... 32
   DIRECTOR Commands ................................................................................ 32
      Accepting and Interpreting Commands ................................................... 33
      Retrieving Text ....................................................................................... 33
      Transmitting Text to CRTS ...................................................................... 33
      Controlling Position of Cursor and Text on CRT Screen ......................... 34
      Controlling Presentation of Materials on Film ......................................... 34

7  PERSPECTIVE ............................................................................................. 35
   Means vs. Ends .......................................................................................... 35
   Media .......................................................................................................... 35
   Further Development of the Text-Handling Subsystem ............................... 36

Literature Cited ............................................................................................. 39

Appendix
A  Summary of Text-Handling Terminology ............................................... 41

Figures
1  Text Handling in Course Development .................................................... 9
2  Text Handling in Course Administration ................................................ 10
3  CRT Display With Question for Student .................................................. 16
4  CRT Display With Student Response and Feedback After Student Response 16
5  Schematic of CRT Block Structure for D1977 ............................................ 17
6  Excerpts From EDITOR Course .............................................................. 19
7  Sample Image Output ................................................................................ 25
8  Sample Dumptext Report ......................................................................... 26
9  Sample String Report .............................................................................. 27
10 Sample Portion of an IMPACT Coursewriter Program ............................ 30
A-1 Text Sets ............................................................................................... 43
Project IMPACT—
Computer-Administered Instruction:
Preparing and Managing the
Content of Instruction,
IMPACT Text-Handling Subsystem
Part 1

INTRODUCTION

BACKGROUND

The objective of Project IMPACT (1) is to evolve a series of prototype systems of computer-administered instruction (CAI) in order to produce a total CAI system that is effective, efficient, and cost-effective for operational use in Army training. The prototype CAI systems developed in Project IMPACT include four kinds of components:

1. Hardware: The computer, student stations, and associated equipment.
2. Software: The computer programs that control operation of the hardware.
3. Courses of Instruction: The content and logic of courses of instruction.
4. Instructional Decision Models (IDMs): The rules and strategies for deciding what instruction to present next to an individual student.

The hardware and software components of the IMPACT system are described in Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem, HumRRO Technical Report 70-22 (2). The present document describes in more detail the characteristics and capabilities of the hardware and computer software that are involved in preparing and managing the content (text) of courses of instruction.

In a developmental project for CAI, such as IMPACT, various approaches to design are explored. The overall goal is to evolve efficient, effective, and economical systems and subsystems that may then be incorporated into an operational environment. The system components described here are a part of a “first-generation” prototype in operation at Project IMPACT but not intended for implementation directly in Army schools.

In computer-administered instruction a great deal of information is transmitted back and forth between the computer and each student. The information provided by the computer system to the student may take many forms, depending on instructional objectives and strategies—narrative explanations, questions and practical exercises, definition of terms, graphic illustrations, words of encouragement or reproof, study assignments, or literature references—in short, any of the kinds of information a tutor might provide his student.

The development of computer-administered courses involves at least two distinct processes, the first of which handles the substance or content of the instruction (i.e., the subject matter). This process includes writing and arranging the content in a way that is appropriate for computer presentation and for interaction between the learner and instructional decision model. It also involves storing this content in computer readable files. The product of the first process is what is referred to in this document as “text.” Preparing, storing, managing, and retrieving this text is called “text handling.”

The second process (not the focus of this report) concerns the procedures or logic by which the content is presented to students, and by which students communicate with the content and the computer. This process is discussed in HumRRO Technical Report 70-22 (2).

---

1 The plan for Project IMPACT is described in the Technical Development Plan, 2 December 1966, and the relationship between IMPACT and the fulfillment of Army Training needs is given in “USCONARC Long Range ADPS Master Plans (Schools),” Headquarters, U.S. Continental Army Command, Director of Management Information Systems, 27 December 1968.
The combination of hardware, computer software, instructional software, communications facilities, CAI personnel, and procedures that together perform text-handling functions is called the text-handling subsystem. This subsystem may be designed and described from several points of view—hardware and communications engineering, software design and implementation, data management technique, and so on. This report does not approach text handling from a hardware or communications engineering viewpoint; rather text handling is conceived and addressed from the viewpoint of instruction developers in an operational CAI system—authors, instructors, and instructional programmers. The kinds of problems discussed include:

1. The major requirements for preparing and maintaining the content of CAI courses.
2. The computer-based tools the IMPACT system provides to assist authors in preparing instructional text.
3. How the IMPACT text-handling subsystem minimizes hardware and software constraints on instructional design and strategies.
4. How the IMPACT text-handling subsystem interacts with other components of the CAI system; whether it is dependent on specific hardware devices, software systems, or CAI languages.
5. What the components of the IMPACT text-handling system include.
6. How the author of a course uses the facilities provided.

TEXT-HANDLING CONSIDERATIONS IN CAI

Some of the major considerations and problems in text handling are:

1. The conversion of text from author to computer storage.
2. The need to relieve authors of technical hardware and software concerns while they are preparing course content.
3. The relation of text handling to programming languages used to specify the logic of a CAI course.
4. The iterative nature of CAI course development.
5. The management of course development activities.
6. The influence of the text-handling subsystem on course design and instructional strategies.

A major concern in CAI course development is that of converting the authors' instructional materials to computer storage. “Computer storage” means that the text is in computer-readable character strings, with special formatting characters depending on the retrieval devices used; the text is in elements that the computer can identify and retrieve at the appropriate time for an individual student.¹ In any computer-based system there are some hardware and software dependent characteristics to the computer-stored text. Hence, some conversion of text must be made before the author can transfer it to computer storage. The steps in this conversion may be performed by the author, by intermediary personnel and programs, or by hardware.

Course authors and researchers generally want to be unencumbered by technical hardware and software considerations, so that they can be free to attend to pedagogical concerns. This is one of the reasons for the development of more than a dozen “CAI languages” in the past few years. These special-purpose computer languages, such as Coursewriter, INFORM, LYRIC, and TUTOR, partly reflect the desire to have increasingly more of the text conversion steps performed by computer rather than man.

¹See Appendix A for definitions of text-handling terms.
Such languages incorporate a number of predefined functions or procedures (e.g., student record keeping, response analysis, data collection, and decision making) in addition to built-in text-handling functions. The advantages and disadvantages of the special-purpose CAI languages as an approach to CAI programming have been widely debated among CAI researchers. Other kinds of programming languages used for CAI include general-purpose compiler languages such as FORTRAN, PL/1, and ALGOL, and interactive languages such as BASIC and APL.

Text handling is only one of several kinds of functions performed in any of these languages. The languages cannot be evaluated on the basis of their efficiency and effectiveness for text handling alone. A language may provide excellent facilities for storing and retrieving text, but not provide the computational or string manipulation facilities required for a particular instructional strategy; or it may provide text-handling facilities suited to a particular instructional strategy but not to others (e.g., tutorial gaming).

By isolating the text-handling functions of a CAI system from other functions performed through software and language, a flexible, generalized text-handling subsystem can be designed without constraining the choice of languages used to program other functions of a CAI system. The languages then used for programming the other functions (e.g., decision making, response analysis, and computation) may be selected on the basis of their applicability to a specific instructional strategy or mode (e.g., simulation, games, student inquiry, tutorial, problem solving).

Another major text-handling consideration derives from the iterative nature of CAI course development. A CAI course is continually refined as course developers obtain feedback on student performance, so that the text undergoes continual modification. Course developers must have a rapid, simple, inexpensive way to retrieve and modify individual elements of text.

A related consideration in text handling is the management of course development activities. For lengthy tutorial courses there are thousands of individual elements of text to be created, converted, and revised. Coordinating and monitoring this effort is complex, particularly where more than one author is involved.

A critical consideration in text handling is the need to minimize constraints on course design and instructional strategy. The way in which the text is stored, identified, and retrieved in the CAI system has tremendous influence on the flexibility the course developers have in design of the course. For example, some special-purpose CAI languages handle text in such a way that it is extremely awkward for the author to provide for student-initiated inquiries. In some cases it is also difficult for the author to re-use or recombine segments of text in different ways for different students. Such constraints allow the author less flexibility in the instructional strategies he can employ. The more such constraints exist, the more the author must be concerned with technical problems, computer hardware, and software techniques rather than with pedagogical techniques.

**OBJECTIVES OF THE IMPACT TEXT-HANDLING SUBSYSTEM**

In light of these considerations, the objectives of the IMPACT text-handling subsystem are to:

1. Minimize costs of course development by providing a fast, easily used way for the authors to create text and have it converted to computer storage.
2. Make the course development process more efficient by providing tools for managing the development process.

1 Examples of the “CAI Language” discussions are found in Zinn (3, 4).
(3) Help ensure the effectiveness of CAI courses by providing rapid, simple means of modifying and maintaining the text.

(4) Help ensure the effectiveness of CAI courses by providing for flexibility of course design and by minimizing hardware/software constraints on course design and instructional strategies.

(5) Provide a generalized capability that will be usable in a variety of operational CAI hardware and software systems.
Part 2

OVERVIEW OF IMPACT TEXT-HANDLING SUBSYSTEM

GENERAL CHARACTERISTICS

The salient characteristic of the text-handling subsystem, which is designed to meet the objectives discussed in Part 1, is that it physically and logically separates the content of a course from the logic or instructional strategies by which content is presented to students (5). The subsystem is designed to be compatible with a variety of different languages that might be used to program the logic, or strategy, of a course. It is also designed to be implemented with a variety of different CRT devices.

In this approach, the course development team is provided direct online access (via CRT) to the computer-stored text, for creating, reviewing, and modifying text elements. Offline access is also provided in the form of hardcopy printouts of text.

Text to be presented via CRT is prepared and stored in the form of small units called elements which may then be "strung together" in various sequences and combinations by course programs according to the desired instructional strategy. Materials presented via film projector are stored on film but retrieved through mechanisms similar to CRT-text retrieval.

The computer programs that contain course logic interact with the text-handling subsystem through commands to a text-handling software supervisor. The choice of the language in which those computer programs are written depends on the requirements of the instructional strategies, the languages available at the CAI installation, and the type of programming expertise available to the development staff.

The IMPACT-A text-handling subsystem supports two basic types of activity in the CAI center—course development and course administration.1

COURSE DEVELOPMENT

The components of the text-handling subsystem that are involved in course development are:

(1) Authors and other members of the course development team.
(2) The CRT device (the Sanders 720, 6), used for communicating with the computer.
(3) EDITOR (Entry on Disk of Instructional Text for Online Retrieval), a simple yet powerful set of text-handling commands used by authors to create, modify, and retrieve text online.
(4) Text Files on magnetic disk or drum at the computer center.
(5) FACS (File Activity Control System), a set of computer programs that provides course developers with information about the status and contents of text files.

1 See The IMPACT Staff. Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem, HumRRO Technical Report 70-22, December 1970 (2) for descriptions of the total hardware-software subsystem as it relates to these activities.
The text-handling subsystem as it is used in course development is shown in Figure 1. Members of the course development team (authors) create, revise, and retrieve text online at the CRT through the use of EDITOR commands. EDITOR interprets the commands and executes them. EDITOR writes the text on (and retrieves it from) the text files. FACS programs provide text documentation for use as worksheets for further course development work such as editing and reports used in managing course development activities. More than one person at a time can perform these activities, on different CRTs.

**COURSE ADMINISTRATION**

The components of the text-handling subsystem that are involved in course administration are:

1. **DIRECTOR**, the text-handling supervisor. DIRECTOR controls the retrieval of text from text files, and transmits it to a student’s CRT. DIRECTOR is the interface between course logic and the text-handling subsystem.

2. The CRT device used to communicate between student and computer.

3. The text files containing the elements of text to be combined in various ways to form instructional units of a course.

4. The Perceptoscope film projector.

The text-handling subsystem as it operates during course administration is shown in Figure 2. Course programs (not considered part of the text-handling subsystem) analyze student responses and make instructional decisions. Based on those decisions, the course programs send commands to DIRECTOR as to which text materials to transmit next to the student. DIRECTOR retrieves the appropriate text from the text files and transmits it to the student’s CRT. DIRECTOR also transmits commands to the Perceptoscope, which selects and projects film presentations.

In the remainder of this document, EDITOR and FACS are described from the point of view of course development activities. DIRECTOR is described from the course administration point of view.
Text Handling in Course Development

ONLINE OPERATIONS

OFFLINE OPERATIONS

Figure 1
Text Handling in Course Administration

Film Projector

CRT

Student

Instructional Text

Student Responses

DIRECTOR

IDM and Course Programs

Computer

Text Files

Course program files

Disk

Disk

Figure 2
EDITOR (Entry on Disk of Instructional Text for Online Retrieval) is a simple yet powerful set of computer commands that simplifies text handling for members of a CAI course development team. Course authors, instructors, and researchers can create, revise, and retrieve text online at the CRT, without going through intermediaries such as CAI language programs, human coders, or computer center procedures. Any element of text may be retrieved directly and viewed on the CRT as it will appear to students. Various sequences of text may be reviewed independently of CAI course programs.

The following is an example of how a CAI team member might use EDITOR. (The commands shown here are explained in detail in later sections.)

A course author (Smith) sits down at the CRT, with notes about the text he wants to compose. His first element of text will be called INTROSERIES. He types on the CRT keyboard the command:

```
CREATE D1977, BY SMITH, INTROSERIES
```

This command tells the computer that Smith is about to create element D1977, titled "Introseries." If he wishes, Smith can use abbreviations:

```
CRE D1977, SMITH, INTROSERIES
```

Smith transmits the command to the computer by pressing the SEND key on the keyboard and then waits a second or two for the computer to tell him to proceed. He then types in his text. He can insert or delete words, lines, or paragraphs of text, or rearrange portions on the screen, through the use of editing keys built into the Sanders 720 CRT. When he is satisfied that the text looks the way he wants it for his students, he SENDS the text element to the computer. The computer stores the element on the text file. Smith receives an acknowledgement that the CREATE is completed.

Smith now wants to see how the INTROSERIES text looks to the student in the context of a series of elements in an instructional unit. He types the command:

```
(DISPLAY D1975), (DIS D1976), (DIS D1977)
```

Element D1975 immediately appears on the CRT screen. Smith reads it and presses SEND. Element D1976 appears on the screen. Smith reads that and again presses SEND. Element D1977, INTROSERIES, appears on the screen. Smith is satisfied with the continuity among the three elements, but wonders how 1977 would look to a student who received a different unit of instruction in which he would see D1977 after D1968. He types the command:

```
(DIS D1968), (DIS D1977)
```

He reads D1968, presses SEND, and then reads D1977. He notices a slight discontinuity from 1968 to 1977, which can be remedied with a minor change in wording on D1968. He types the command:

```
MODIFY D1968, SMITH
```

See Figure 3 for an example of how the text will look on the CRT.
D1968 appears on the screen again. Smith changes the last sentence in D1968 and sends the revised version to the computer, which stores it in place of the old one. Smith receives an acknowledgement that the MODIFY is complete.

EDITOR commands are used by course authors, instructors, editors, and researchers—in short, anyone who has anything to do with the text of a course. The commands are used to create new text, and to modify existing text. They are used to retrieve selected text elements, or to review various sequences of instructional frames. EDITOR commands may be used to extract text sequences from a course for demonstration or research purposes.

Depending on the organization of the CAI project and course development staff, the EDITOR commands will be used in different activities by different persons. For example, in a large course production effort, a separate clerical staff may be assigned the job of inputting the text on the CRT. In this case, they would work from specifications from the course authors, showing how the text is to be formatted on the screen (spacing, columns, etc.). The course authors might then use EDITOR commands to review that work to see how it looks on the screen.

Depending on his capabilities and interests, and the standard operating procedures of the staff, the course author may choose to create the text online himself without the assistance of intermediate personnel. It is particularly useful to the author when he is experimenting with new techniques for presenting ideas on the CRT, to be able to see immediately how the material will look to the student.

THE SIX EDITOR COMMANDS

There are six EDITOR commands, beginning with the following verbs or their abbreviations:

CREATE (CRE)
MODIFY (MOD)
DELETE (DEL)
COPY (COP)
DISPLAY (DIS)
ROLL ON (ROL)

EDITOR commands operate on individual text elements. An element is a uniquely identified unit of text for storage and retrieval purposes, which may be as short as one line or as long as an entire CRT page. Elements are the basic modules of information that can be combined and recombined in various ways to present instruction to the student. Such a combination of elements is called a page.

The EDITOR commands include the verb (e.g., CREATE), and the identifier for the element being operated on (e.g., D37), and in some cases additional information about the element, (e.g., title). The identifier and the additional information are called the arguments of the command.

These commands may all be initiated at the CRT keyboard, by typing the verb and its arguments, and enclosing the command in parentheses. The verb may be spelled out fully, or abbreviated by using the first three letters, and arguments are separated by commas. The uses of these commands are described in the following sections.
CREATE (CRE)

Purpose. CREATE is used to store a new element on a text file.

Format. (CREATE identifier, name, title).
- **Identifier** is a code that indicates which element within which text file is to be created. These codes are assigned by the course development team (See page 11 for a discussion of the text file organization).
- **Name** is the name of the person creating the element.
- **Title** is the title of the element. Titles can be used to provide more meaningful information to the CAI team than the coded (element) identifiers.
- The entire command is enclosed in parentheses. The arguments are separated by commas.

Use. To CREATE a new element:
- Type the command on the CRT keyboard, e.g. (CRE D1977, SMITH, INTRO-SERIES)
- Press SEND BLOCK key and wait for the computer to acknowledge the request.
- CLEAR the screen (press CLEAR key).
- Type the text of the element. After it looks the way you want it—
- SEND PAGE. After the computer has completed the CREATE, an acknowledgment appears on the screen. This is the case with all commands.

MODIFY (MOD)

Purpose. MODIFY is used to make changes to an element that has already been stored.

Format. (MODIFY identifier, name, new-title).
- **Identifier** is the code by which an element was previously stored.
- **Name** is the name of the person making the change.
- **New-title** is used if the title is to be changed from the old one. Otherwise title may be omitted from the command.

Use. To MODIFY an existing element:
- Type the command on the CRT keyboard, e.g., (MOD D1977, JONES, INTRO-SERIES2).
- SEND the command to the computer (Press SEND BLOCK key).
- Wait for the computer to transmit the old D1977 to the screen (this usually takes about half a second).
- When element D1977 appears on the screen, make the desired changes by replacing, deleting, or inserting characters, words, or lines of text. The keyboard has special editing keys that make it easy to change parts of the text without retyping the entire segment. When all desired changes have been made—
- SEND the revised element to the computer (Press SEND PAGE key).

DELETE (DEL)

Purpose. DELETE is used to delete an element from the file.

Format. (DELETE identifier, name).
- **Identifier** is the code under which the element was created.
- **Name** is the name of the person making the deletion.
Use. To DELETE an existing element simply type the command, e.g. **(DELETE D1977, HARRY)**, and press SEND BLOCK key.

**COPY (COP)**

**Purpose.** COPY is used to copy the text of an element that has already been stored into another place on the display files. This capability may be exercised when several different versions of the same basic element are desired. An element can be copied into several different file locations (the MODIFY command may then be used on the copied elements, to make the alterations required for the different versions) or it can be copied to another text file.

**Format.** **(COPY from-identifier, to-identifier, name, new-title).**
- **From-identifier** is the key under which the element was originally stored.
- **To-identifier** is the key to which the element is to be copied.
- **Name** is the name of the person initiating the COPY.
- **New-title** is the title to be assigned to the new copy of the element. If no new title is desired, omit the argument.

**Use.** To COPY an existing element simply type the command, e.g., **(COPY D1977, D1980, JANE, NEWSERIES)**, and press SEND BLOCK.

**DISPLAY (DIS)**

**Purpose.** The DISPLAY command is used to retrieve prestored elements from the files and view them on the CRT. This command makes it possible for any member of the course development team to view any element or series of elements of the course at any time. One can “step through” a series of instructional elements to see how they will look to a student, or for demonstration purposes.

**Format.** **(DISPLAY identifier).**

**Use.** To call an element to the CRT screen, simply type in the command, e.g. **(DIS D1977)**, and press SEND BLOCK. The computer clears the screen and then transmits the segment to the screen.

**ROLL ON (ROL)**

**Purpose.** The ROLL ON command is used to retrieve a prestored element from the files for viewing on the CRT, as with DISPLAY. ROLL ON differs from DISPLAY in that the element is rolled on to a specified position on the screen, without disturbing what is already on the screen, whereas DISPLAY first clears the screen. The format and use of ROLL ON are described on page 17 following the discussion of blocking.
ROLL ONS AND BLOCKING

One of the greatest advantages of CAI from an author's standpoint is that it can provide continuous interaction between an individual student and his computer-tutor. The computer can elicit frequent responses from the student, who can get immediate feedback in the form of special instruction, test scores, answers to his inquiries, and so on. The CRT is particularly useful for this kind of interaction, because a conversation can take place on the screen as though the student and computer were writing notes to one another on a chalkboard or scratch pad. The easier it is for the author to specify the nature of the conversation that is to take place, the more flexible and responsive the instructional system can be to the individual student.

In Figures 3 and 4, a conversation between student and computer is illustrated. In Figure 3 the student sees a question asked of him. His own answer, and the computer's feedback to him concerning his answer are shown in Figure 4. When the feedback is transmitted to the screen the original question and the student's answer are left undisturbed. This technique is called a "roll on," because, in effect, an element of text is "rolled on" to the screen.

In order to roll a text element onto the right place on the screen, the computer needs a way of addressing specific portions of the screen. In some display systems this is achieved by addressing the actual coordinates of the screen—for example, 10 lines down, seven spaces across.

BLOCKING

In the Sanders 720 display system screen positions are addressed through a mechanism called "blocking." EDITOR has the capability of addressing these blocks. The text elements have embedded in them special characters that indicate the beginning of a new block. The blocks can then be addressed by number. So, for example, the feedback (as shown in Figure 4) was rolled on to Block 3. The spaces where the student gave his answers are a separate block, Block 2. When the student transmits his answer to the computer he transmits only the answer block (by pressing SEND BLOCK key on the keyboard). Other text on the screen remains unaffected by that transmission. This minimizes communications line usage.

Hence again the authors need not specify the screen coordinates of student responses. They simply create the text element with the appropriate blocking characters embedded in them. For example, D1977 (Figure 4) contains three blocks—one for the question, one for the student's answer, and one for the feedback roll on. The text of the feedback itself was created as a separate element and stored on the text file with its own identifier, F105. After the student sent his answer to the computer, F105 was rolled onto the screen. Had he given a different answer, a different feedback would have been rolled on.

A schematic of the block structure for D1977 is shown in Figure 5.
THE ROLL ON COMMAND

Purpose. ROLL ON is used to retrieve a selected element for viewing on the CRT screen. The element is rolled on to a specified block on the screen.

Format. (ROLL ON identifier, block-number)
- **Identifier** is the key for the element to be rolled on, e.g., F105.
- **Block number** is the number of the block that the element is to be rolled onto, e.g., (block) 3.

Use. A typical sequence for using ROLL ON is as follows:
- Use DISPLAY command to retrieve a master display for viewing on the screen. That master display should contain a blank block for roll ons.
- Without clearing the screen, type in the ROLL ON command for the element to be rolled on, e.g., (ROL F105, 3). Press SEND BLOCK.
- The new element will appear on the screen in the block requested. That element can then be seen as it will appear to the student, in conjunction with the master display.

MULTIPLE COMMANDS

In the example on page 11, Smith first issued the command to CREATE element D1977. After he created D1977, he issued a command to DISPLAY D1975, D1976, and D1977. Instead of issuing two separate commands, however, he could have issued the
following command at the outset: \((\text{CREATE D1977, SMITH, INTROSERIES})(\text{DIS 1975})(\text{DIS 1976})(\text{DIS 1977})\). Had he done this, the computer would have automatically transmitted D1975 to the screen as soon as Smith was finished creating D1977.

This capability for issuing multiple commands in one statement has a variety of uses. For example, if Smith wanted four slightly different versions of D1977 for his course, he would first have the original D1977 copied three times by issuing the following command: \((\text{COPY D1977, D1981, SMITH, INTROSERIES2})(\text{COPY, D1984})(\text{COPY, F408})\). Then he would call out the elements and alter them, through the \text{MODIFY} command.

This series of commands will \text{COPY} D1977 into D1981, D1984, and F408. Note that the \text{FROM} argument D1977 is \text{omitted} on the second two commands, and only a comma is given to indicate the omitted argument. This is possible because \text{EDITOR} will assume that the later commands are still referencing the argument given on the first command. The same is true of SMITH and INTROSERIES2. All three of the \text{COPIED} versions will have the same title—Introseries2. If different titles were desired for the different versions, the command series would look like this:

\[(\text{COPY D1977, D1981, SMITH, INTROSERIES2})
(\text{COPY, D1984, OTHERTITLE})(\text{COPY, F408, THIRDTITLE}).\]

Note that the \text{FROM} and \text{NAME} arguments D1977 and SMITH, may be omitted in the second two commands, by simply typing the commas that separate the omitted arguments.

Another example of a multiple command is:

\[(\text{COPY D111, 173, SMITH, PRACTICE9})(\text{MODIFY 173}).\]

The above command series will \text{COPY} D111 into D173, give D173 the title PRACTICE9, and project D173 onto the screen. D173 may then be changed. \text{SEND PAGE} will transmit the revised version of D173 to the computer.

**PREPARING A SERIES OF DISPLAYS FOR ONLINE PRESENTATION**

\text{EDITOR} commands may be used to create simple series of displays that may be presented online automatically, without using any other CAI programming language. The series of displays may be a linear or a simple multiple-choice branching course. Such sequences are created by embedding \text{EDITOR} commands into the text of the displays. Thus, one display page automatically calls the next. After the student reads one display he presses the \text{SEND} button, which transmits the element back to the computer. The \text{EDITOR} command embedded in that page then causes the computer to display the next page.

This “page turning” use of \text{EDITOR} commands is very convenient for creating short orientation programs. For example, selected elements from a longer, more complex course may be \text{COPIED} and modified slightly to incorporate the \text{EDITOR} commands into the text. Then the extracted version of the course may be used for orientation or for experiments.

One course created in this way at Project IMPACT is on the subject of using \text{EDITOR} commands. The computer is not programmed to analyze student responses in this course; however, it provides a structured text presentation and enables the student to become familiar with the CRT screen and keyboard. After he steps through the instructional text online he can then practice using the \text{EDITOR} commands he has learned. Sample pages from this course are shown in Figure 6. The \text{EDITOR} commands embedded in the text are circled on the figures.
Excerpts From EDITOR Course

GOOD AFTERNOON

WELCOME TO PROJECT IMPACT

THIS SEQUENCE OF DISPLAYS WILL SHOW YOU HOW TO INPUT, RETRIEVE, AND CHANGE COURSE MATERIAL WHICH IS KEPT IN DISK STORAGE DOWNSTAIRS IN THE COMPUTER ROOM.

TO SEE THE NEXT DISPLAY, PRESS THE "SEND BLOCK" KEY WHICH YOU WILL FIND ON THE LOWER RIGHT SIDE OF THE KEYPAD.

THE COMMANDS WHICH WE USE TO DO THIS CAN BE DIVIDED INTO TWO GROUPS.

ONE OF THESE GROUPS CONSISTS OF THE COMMANDS WHICH WE USE TO CREATE, EDIT AND CHANGE COURSE MATERIAL.

IN THE OTHER GROUP ARE THOSE COMMANDS WHICH WE USE SIMPLY TO CALL COURSE MATERIAL FROM ITS DISK STORAGE LOCATION AND TO PUT IT UP ON THIS SCREEN. LET'S TAKE A LOOK AT HOW WE USE ONE OF THE SECOND GROUP OF COMMANDS.

PRESS SEND BLOCK WHEN READY TO GO ON.

OUR MOST IMPORTANT COMMAND FOR MOVING COURSE MATERIAL FROM DISK STORAGE TO THE SCREEN IS THE "DISPLAY" COMMAND. THE FORMAT OF THE DISPLAY COMMAND IS THIS:

DISPLAY XXX

WHERE XXX CAN BE ANY NUMBER FROM THE COLUMN ENTITLED "DISPLAY" ON PAGE 10 OF YOUR MANUAL.

IF YOU LOOK DOWN TO THE BOTTOM RIGHT CORNER OF THE SCREEN, YOU WILL NOTICE THAT THE CURSOR (THE BLINKING UNDERLINE) IS ON ONE OF THESE DISPLAY COMMANDS. WHEN YOU NEXT PRESS THE SEND BLOCK BUTTON, THE CRT WILL SEND THIS COMMAND TO THE COMPUTER SO THAT YOU CAN SEE THE NEXT DISPLAY.

PRESS SEND BLOCK.

Figure 6
EDITOR commands may also be used to tie together a series of display pages that form an instructional unit. Each page of text automatically calls the next, within the instructional unit. The selection of the unit to be presented to a student at a given point in his instruction will be determined by the course logic program.

THE SIGNIFICANCE OF EDITOR

EDITOR, a key capability in the IMPACT CAI system, has the following significant features:

(1) EDITOR allows persons who are completely naive with respect to computer programming to store and retrieve display materials from computer storage.

(2) EDITOR commands are so simple to use that they may be taught very quickly to untrained personnel.

(3) EDITOR makes it possible to revise instructional materials without completely recreating them.

(4) EDITOR makes it possible for any member of the course development team to create, modify, or simply view any of the instructional display material. Any team member has rapid access to the material at any time, and can view the material on the CRT screen as it will appear to the student.

(5) Text materials may be created, stored, retrieved, and revised via EDITOR, in any sequence at any time. The instructional strategy by which the text is to be viewed by students in a course does not affect the manipulation of text via EDITOR. EDITOR commands may be used to retrieve a series of text elements for various course development and research purposes.

(6) EDITOR makes it possible to prepare, store, edit, review, and revise instructional text without affecting computer programs. This is in contrast to CAI software systems in which instructional text is embedded in the logic instructions of the computer programs. With EDITOR, all text is physically segregated from program steps.
Part 4

THE TEXT FILES

A computer has access to files of material stored on magnetic disk or drum at the computer center. Text that is to be presented to students via CRT is stored on these files.

The IMPACT text-handling subsystem is designed to store, retrieve, and combine elements from any number of different files. The names, purposes, and characteristics of the elements in those files are determined by the course development team to suit the purposes of the CAI installation or an individual author.

MULTIPLE FILES

It is often useful, during course development, to categorize text elements according to their instructional purpose. For example, a separate category of elements might be made of questions to be asked of the student. The different categories of elements also have different physical characteristics—for example, questions are usually relatively short whereas a mainline explanation of a concept may be as long as a whole CRT page.

The different categories of text elements may be stored on different text files.

PRESENT IMPACT TEXT FILES

At the present time Project IMPACT has four text files: "D", "I", "F", and "Q". The characteristics of these files are described as follows:

"D" File. The D file is used for what are called "mainline" or "master" text elements. These are usually long elements containing explanatory text, questions to the student, answer blocks, blank blocks for roll ons, and so on. The mainline instruction in a tutorial course is usually contained in these D elements.

D elements may contain up to 1023 characters (one less than the maximum number of characters that can be displayed on the Sanders 720 CRT screen). The retrieval ID for D elements is the file location number (e.g., D1977).

"I" File. The I (Inquiry) file is used for definitions and review material that a student or author may retrieve by a symbolic name. The maximum length of an I element is 699 characters. The retrieval ID for I elements is a unique string of up to 40 characters.

"F" File. The F file is used for feedback and associative elements, such as an explanation of why a student's answer was wrong, the correct answer to a question, or a short review. They are usually, but not necessarily, used as roll ons.

F elements may be up to 699 characters long. The retrieval ID is the file location (e.g., F37).

"Q" File. The Q file is reserved for questions and problem statements that are rolled onto a master element. Qs, like Fs, are up to 699 characters long and are retrieved by file location ID (e.g., Q11).
RELATION OF TEXT FILES TO COURSES

Text files are not physically or logically tied to specific courses of instruction. Course programs may call on any text elements in any of the files. In addition, elements for one or several courses may be stored on the same files. Some standard messages might be used many times in several different courses. For example, messages found to be useful in specific circumstances in motivating in one course could be called on as needed in other courses. Short orientation courses may call on elements from operational courses. Different versions of a course may be prepared for experimental purposes, all calling on the same text elements.

INFORMATION IN THE FILES

In addition to the actual text, other information is included with each element in a file. This information is used for administrative purposes (it is not presented to the student) for course development and maintenance, and includes the following items:

1. Retrieval ID
2. Title of the element (up to 32 characters)
3. Author name (up to 15 characters)
4. Last action performed on this element (CREATE, MODIFY, DELETE, COPY)
5. Date and time of last action performed on the element

The use of this information is discussed in Part 5 of this report.

RETRIEVAL IDENTIFIERS

Each element on the files has a unique retrieval identifier. Usually, the identifier consists of the actual file location (e.g., D1977 is the identifier for the element at location 1977 in File "D"). In some cases, a symbolic name may be substituted for the file location identifier. Use of a symbolic name enables students or authors to retrieve information directly from the files. This is useful for student inquiries into a glossary or dictionary, for example. In such a case the glossary term would become the symbolic name.

For example, in a course prepared by Project IMPACT, a student may type the following request:

INFO-DATA PROCESSING

"Data Processing" is the symbolic name of an element that contains the definition of data processing. Hence the definition of data processing is rolled onto the screen in response to the student’s request.

FILE INTEGRITY AND SECURITY

If operational requirements so dictate, the text files can be protected from unauthorized changes, by using the author name as a password when modifying or deleting file elements.

The protection of classified instructional materials from unauthorized access, either online or offline, would be provided to operational installations as an optional system feature. The inclusion of hardware, software, and procedural mechanisms to satisfy security requirements will be determined by the classification requirements of the instructional materials and environment.
Part 5

FACS (FILE ACTIVITY CONTROL SYSTEM)

FACS is an automated tool that assists in developing and maintaining text. As shown in Figure 1, FACS is used after text has been created and stored on text files. FACS computer programs read the text files and produce reports concerning the files. Any particular report can be produced on request of the course development team. The primary reports include:

1. **Image**, which presents a text element exactly as it will appear on the CRT, plus information about the element.
2. **Dumptext**, which presents a series of text elements in the order in which they will appear in an instructional unit.
3. **String**, which lists all the text elements containing a particular word or sequence of characters (i.e., a character string).
4. **List and Totals**, administrative reports on the status of files and text elements.

These reports are used in managing the course development effort and in developing and maintaining text. The ways in which the reports are used are discussed in the following section, after which the major reports are described.

**FACS MANAGEMENT USES**

FACS reports provide the manager of course development with the tools for monitoring course development progress and evaluating text production. FACS provides summaries of the text elements that have been created or modified within a particular time period, so that the amount of production can be easily ascertained. Up-to-date lists of text elements by title and author give the manager a ready reference to the individual who has worked on specific text elements.

Recent modifications to the course come to the manager's attention through selective Image and Dumptext reports, which show text elements changed since a particular date.

The manager can review all the text related to a particular course objective through Image or Dumptext reports that have selected those elements. He can review all text related to a specific subject with the help of String reports that list all text containing a particular word or phrase.

The manager can maintain a record by file of the records that are in use.

**FACS TEXT DEVELOPMENT USES**

Members of the course development team have many occasions to refer to previously stored text. They may retrieve the text online via the CRT and EDITOR commands. For certain tasks it is more efficient to work from hard-copy reports, and FACS makes it possible to maintain current hard-copy documentation of all the text at all times. These copies become worksheets for the members of the team. For example, an author may note corrections and changes on an Image report. The worksheet then becomes input to a clerical assistant who inputs the changes to the computer. Once the change is made, FACS provides
an updated version of the text to keep in the documentation. Given the iterative nature of CAI text development, it would be nearly impossible to maintain an up-to-date documentation manually.

FACS administrative reports assist the members of the development team by providing an up-to-date log of all text file locations that have been used, and the locations available, as well as the titles and authors of each element in the files. Thus, the course development team is relieved of the tedious task of maintaining activity and status logs on the text.

TEXT DOCUMENTATION AND WORKSHEETS—THE IMAGE REPORTS

Image is the principal FACS output. It contains a replica of a text element, as it appears on the CRT screen. Image also shows the characters of the element exactly as they appear in computer storage.

Figure 7 shows a sample Image report page. The nonshaded portion at the top of the page provides administrative information about the text element: its identifying number; its title; the last operation performed on the element; the date and time the operation was performed; and the author's name. The light shaded middle portion is the image of the text as it appears on the CRT. The dark shaded portion provides technical information about the text, including an image of the way it appears in computer storage.

When an Image report is requested, the user can specify which text elements he wishes to have included in the report. He has the following options:

1. Include all elements within a specified range in a particular text file (e.g., D1100 through D1275).
2. Include only elements that were created, modified, or copied since the last Image report.
3. Include all elements in a particular file.

As discussed on page 23, Image reports are used in a variety of ways as documentation and worksheets by course developers.

A SERIES OF TEXT ELEMENTS—THE DUMPTEXT REPORTS

Dumpertext reports provide an overview of a series of text elements. As shown in Figure 8, a series of text elements from different files, which forms an instructional unit, is condensed on one report page. In the example in Figure 8, a mainline element, D1105, is shown (circled) on the left. Note that it is printed in abbreviated form, without the spacing and formatting that would be seen on the CRT screen. In the middle of the page is a question element, Q503. This is a question that will be rolled on the screen with D1105. On the right are three “feedback” elements, F552, F553, and F554. These are elements that will be rolled on the screen, depending on the student's response to the question.

Dumpertext reports are used in a number of different ways by the course development team. They are used to get a quick overview of a series of related text elements, or to check on continuity and consistency. This is particularly important when several authors have written different portions of the text. Some authors use Dumpertext reports as worksheets or flowcharts for specifying course logic to the course programmers. The condition under which each element is rolled onto the screen may be handwritten on the report page next to the printed text.
Complete the sentences below with the following words:

COBOL  COMPUTER  DATA PROCESSING  PROGRAM

Give a show of confidence after each answer. Press send after each answer.

1. Handling large amounts of information is called _______________.
2. A machine that can process data rapidly is a _______________.
3. A series of steps for directing computers is a _______________.
4. The programs you write will be in the language _______________.

Press send.
Sample Dumptext Report

PROBLEM 1 SPECIFICATION (SPEC) INPUT: A FILE OF RECORDS (PUNCHED CARDS) CONTAINING THE NAMES AND SERIAL NUMBERS OF PERSONNEL. OUTPUT: A FILE OF RECORDS (PRINTED LIST) CONTAINING THE DATA IN EACH INPUT RECORD. PREPARE A COBOL PROGRAM THAT ENABLES THE COMPUTER SYSTEM TO READ THE RECORDS IN THE INPUT FILE, USING THE CARD-READER, AND WRITE THE DATA, USING THE PRINTER.

Figure 8
AN ANALYTICAL TOOL—THE STRING REPORTS

String reports are an analytical tool used to study course content for a variety of purposes. The String program searches through the text files to find character strings specified by the user. Figure 9 shows a sample String report. "SYSPUNCH" (①) is a character string specified by the user. The report lists the elements (②) that SYSPUNCH appears in, and the number of times SYSPUNCH appears in each element (③). The total number of display elements SYSPUNCH appears in is summarized (④), and the total number of times SYSPUNCH appears in those elements is shown (⑤). The sample report includes similar information on the character string "CONSOLE".

Sample String Report

FILE D:
① SYSPUNCH APPEARS IN:
③ COUNT OF OCCURRENCES = 2
④ COUNT OF DISPLAYS = 2
⑤ COUNT OF OCCURRENCES = 2

FILE D:
CONSOLE APPEARS IN:
③ COUNT OF OCCURRENCES = 3
④ COUNT OF DISPLAYS = 3
⑤ TOTAL NUMBER OF DISPLAYS CONTAINING STRINGS IS 5

Examples of the ways in which String has been used include:

1. In developing a glossary for a course, a team member asked for String reports on 100 words or phrases that were candidates for the glossary. On the basis of the String reports, the 25 most frequently used terms were selected.

2. To check on the consistency of the use of several terms, the manager of course development requested a String report on these terms. With the String report, he was able to scan the selected text elements to check on consistent use of the same term throughout the course.

3. After several experimental students had taken a course, it was determined that two words in the course were being used in ways that confused the students, and it was decided to replace them with a new phrase. String reports enabled course authors to identify the offending text elements and make the appropriate modifications.

ADMINISTRATIVE REPORTS

Two FACS reports provide administrative logs of file and text status. The List report is simply a list of all active file locations, showing the element identifier and title for that file location. It provides an up-to-date reference for course developers of the status of the text files.

The Totals report shows at a glance the production that has taken place in the text development effort. For each text file, the report shows the number of elements created, modified, copied, and deleted since the last report date.
HOW THE COMPUTER HANDLES TEXT
DURING COURSE ADMINISTRATION

DIRECTOR (Directing Text for Online Retrieval) plays a key role in course administration. When a student is taking a course, DIRECTOR guides the text that the student sees on the CRT. The IDM and course programs tell DIRECTOR what combinations of text elements to display to a particular student. DIRECTOR receives the commands from the course programs, retrieves the appropriate text elements from the files, and transmits them to the appropriate place on the student's CRT screen.

DIRECTOR commands are very similar to EDITOR commands, with two exceptions:

1. While EDITOR commands are initiated by a human online at the CRT keyboard, DIRECTOR commands are normally initiated by the computer programs containing the logic of a course.
2. DIRECTOR commands contain certain features and options not included in EDITOR. These provide flexibility in text handling during course administration, and efficiency in course program implementation.

DIRECTOR interfaces with course programs and CRT hardware devices. The relation of DIRECTOR to those components is explained in the following section.

DIRECTOR AND COURSE PROGRAMS

Course programs are computer programs that contain the logic of a course—that is, the rules for analyzing student responses, the rules for choosing the instruction to present next, instructions to collect data on student performance, and so on. At Project IMPACT, these course programs are usually written in the IMPACT-Coursewriter language, since Coursewriter's computer software has built into it many such useful capabilities as keeping student records, analyzing student responses, and collecting data on student performance.¹

DIRECTOR, however, like the other components of the text-handling subsystem, is not dependent on Coursewriter. Text-handling commands may be issued to DIRECTOR from computer programs that are written in any language suitable for expressing the course logic.

By analogy, the course program with the IDM may be thought of as the professor and DIRECTOR as his assistant. The professor plans the instruction and conducts the class session, while the assistant helps get the material ready and distributes it to the students.

The interface between course programs and DIRECTOR is as follows: When a student is taking a course, the course program determines what text the student needs

¹IMPACT-Coursewriter is an extension of Coursewriter III for the IBM 360 computer. Coursewriter III is the special purpose CAI language developed by IBM for use on the 360 computer (Coursewriter I and II were for the 1401 and the 1500 computers). Project IMPACT has extended the capabilities of Coursewriter III to support CRT terminals and to make course programs more efficient.
The course program then issues an instruction that causes the material to be output to the student.

For the readers familiar with Coursewriter the following example is provided:

In the case of Coursewriter, the instruction takes the form of a `qu`, `ty`, or `rd`. They are standard output instructions built into the Coursewriter language. In Coursewriter III, without DIRECTOR, the text to be transmitted to the student would be included in the content of the output instruction, for example:

```
qu What is 2 plus 2?
```

The computer would output the character string "What is 2 plus 2?" to the student station.

In IMPACT-Coursewriter with DIRECTOR a command replaces the actual text in the instruction, for example:

```
qu (DISPLAY D1100)
```

When this `qu` instruction is executed, DIRECTOR intercepts the command "(DISPLAY D1100)." DIRECTOR then interprets and executes the command to retrieve and transmit the text element D1100 to the student.

Similarly, course programs written in languages other than IMPACT-Coursewriter may issue output instructions containing DIRECTOR commands. DIRECTOR intercepts the output instruction to see whether it contains a DIRECTOR command. If it does, DIRECTOR executes the command. DIRECTOR may operate with more than one course program at a time, and with more than one programming language at a time. For example, the instructional programming staff may choose PL-1 as a language for programming a particular course. The language offers powerful data manipulation and string handling capabilities useful for complex response analysis. Any other computer language that operates under the IBM 360 operating system may interface with DIRECTOR.

**DIRECTOR AND CRT HARDWARE**

The CRT device used at Project IMPACT is the Sanders 720 and keyboard. This device has certain characteristics that make it attractive for CAI applications (e.g., the blocking capability described in Part 3 of this report). DIRECTOR commands are designed to take full advantage of the capabilities of this particular device.

However, DIRECTOR (as well as the other components of the text-handling subsystem) is not dependent on the Sanders 720. That is, the computer software for DIRECTOR was designed in such a way that it may be implemented for use with CRT devices other than the Sanders 720, even if they have features and capabilities that differ from the Sanders 720. If an alphanumeric CRT device is compatible with the 360 computer and 360 Operating System, it is also compatible with DIRECTOR; however, support for other CRT's is not currently implemented at IMPACT.

**DIRECTOR AND COURSE AUTHORS**

The text-handling commands are easy to learn and use by persons with no knowledge or experience in computer programming. If an author codes the course programs, he can easily learn to incorporate DIRECTOR commands into those programs.

Some CAI projects are organized and staffed so that the course author himself does not write the programs, but rather supplies logic specifications to a computer programmer. In this case, the author does not learn all the details of DIRECTOR text-handling commands. He needs to know, aside from the EDITOR commands discussed in Part 3 of this report, only the capabilities of DIRECTOR and how they affect his course. This section describes these functional capabilities.
The primary capabilities from an author's viewpoint are four: Dynamic text creation, conversational interaction, student inquiry, and generalized course logic.

**DYNAMIC TEXT CREATION**

The phrase "dynamic text creation" used here refers to the creation of pages of text by DIRECTOR, at the time a student is online. (DIRECTOR does not actually "create" the text in the sense that a writer does offline or which an author might do online using EDITOR commands.) DIRECTOR combines text elements to form a CRT page. The pages of text are created by combining text from one or more of the following sources:

1. Elements from the various text files (described in Part 4).
2. Material generated in real time by the course programs (e.g., test scores, or diagnostic summaries for the individual student).
3. Material input by the student himself (e.g., his actual response to a question, or his questions to the computer).
4. Data from buffers and counters in the course logic section.

**Sample Portion of an IMPACT Coursewriter Program**

```
DXP40
  1- 0 RD (DIS D 365, 1)  (Command to DIRECTOR to transmit display)
  1- 1 EP                (Command to poll student for response)
  1- 2 FN RECORD         (Store a copy of the student's record)

DBP22
  1- 0 QU (ROL Q260, 1, 2) (Command to DIRECTOR to transmit question)
  1- 1 EP                (Poll student for response)
  1- 2 ED CH*/CH*/CH*    (Instructions for recognizing student's response)
  1- 3 LD DBP24/R1       "
  1- 4 LD 33/C69         "
  1- 5 AA (L) WORKING*STORAGE* & "
  1- 6 LD 1/S1           "
  1- 7 LD 4/B1,52,1      (Load decision factor values into student's record)
  1-1- 8 BR VAL02        (Transfer control to IDM)
```

Figure 10
Dynamic text creation, from an author's point of view, means that:

1. He can write a text element once and have it used and reused in different ways in different parts of a course (e.g., standard feedback messages, reminder messages, review summaries, or standard remedial material).

2. He can have information that is pertinent to a particular student at a given point in time inserted into standard prestored text elements. For a simple example, the program might insert the student's name into a prestored element.

3. He can have material the student himself input, combined with prestored text. For example, as shown in Figure 10, the author can ask a student to rethink an answer that he, the student, had given earlier in the course.

4. The author does not need to write every unique page of text that will be seen by every student in the course. He writes only the text elements; and, in conjunction with the modeling (IDM) team members, specifies the rules for combining them into pages.

CONVERSATIONAL INTERACTION

If student and computer are to carry on a conversation via the CRT, the CRT device itself must be controlled in a number of ways. For example, when a student answers a question, it is often desirable to retain the question on the CRT while the student is answering. Therefore, his answer must be written on the screen in a particular place, so that the question is not destroyed. When the feedback from the computer is transmitted to the student, in order for him to compare his answer with the feedback, the additional text should not interfere with his original answer. When a student requests the definition of a term, the definition should appear on the screen in such a way that the text he was responding to is not destroyed. When he has finished reading the definition, it should be erased from the screen but the original text should remain intact.

The software capabilities for manipulating the CRT can greatly affect the versatility and smoothness of the "conversational" nature of a system. DIRECTOR is designed to assist both authors and students in this manipulation of the CRT, which is effected primarily through cursor manipulation. The cursor is the small blinking spot on the CRT screen that indicates where the next character will be written. DIRECTOR controls the positioning of the cursor, so that text will be written on the appropriate place on the screen. This occurs both when the computer is writing on the screen and when the student is writing on the screen (through the keyboard). Cursor manipulation, combined with the blocking feature discussed in Part 3 of this report, make it possible to use the CRT in a "scratch pad" fashion, in which portions of the text are erased while others remain, and small portions of text are transmitted back and forth between student and computer using the "scratch pad." The student may also use the electronic pen on the CRT to move the cursor about on the screen. He does this by touching the pen to the spot on the screen where he wants to type.

STUDENT INQUIRY

As stated earlier, DIRECTOR commands are initiated by course programs, and therefore the course programs have control over what text is presented to each student. It is also possible for a student to initiate a command. For example, a student may ask for the definition of a term by typing a special, designated character or group of characters. For example, in the IMPACT-A system the student types "INFO" and the term he wants defined. He may type "INFO-COBOL," asking for a definition of
“COBOL.” DIRECTOR interprets the request as a command to DISPLAY the text element from the Inquiry file that shows the definition of COBOL. The information to be made available upon student request, and the retrieval keys used by the student to access this information, are entirely at the author's option. For example, in Project IMPACT'S course on COBOL, "INFO-" is preformatted on the student's CRT page during those portions of the course where the inquiry capability is to be made available to him. Thus, the student needs to type only the term he wants defined. Retrieval hierarchies may be designed by the author, in which the student retrieves successively more detail on a particular topic through successive inquiries.

One example of the application of these capabilities might be in the case of “hands on” practical exercises in which the student requests instruction or conceptual materials on an “as needed” basis, in support of work with equipment.

Course programs may exercise control over when the inquiry capability is available to the student. Thus, for example, during end-of-course criterion tests the inquiry capability may be turned off. If the student makes an inquiry, DIRECTOR tells him that the request is disallowed.

The course author may provide for a motion or single frame film presentation in response to a student's inquiry. For example, when the student types "INFO-COBOL", he might receive a motion picture presentation on the subject, in addition to the CRT text presentation.

**GENERALIZED COURSE LOGIC**

At Project IMPACT, most of the logic of a course is incorporated in a generalized Instructional Decision Model (IDM, 5). This IDM contains rules for determining what text to present next to a student, on the basis of certain factors about his performance. The IDM program is designed in such a way that the logic routines (which include DIRECTOR commands) are used repeatedly, that is, they are generalized logic subroutines. Hence, course authors and researchers specify the general logic, and special deviations from it. They do not need to specify exactly what decision to make at each point in the course for each student. An example of a general rule is, “If the student is in Knowledge State 2 following a question of Type A, then provide him with feedback Type Y.”

Certain characteristics of DIRECTOR make it possible to implement such generalized course logic. The course program can determine the text element to be presented next through the use of a mathematical algorithm.

For example, the specific Type Y associative statement feedback to be presented might be computed on the basis of an algorithm that relates question types, question number, feedback types (e.g., confirming feedback, review, or other remediation), and file locations assigned to particular feedback types. The results of the computation are stored in a specified place in computer storage. DIRECTOR will treat this place as containing the identifier for the next element to be retrieved from the text files, when given a command that contains a symbolic variable. The ability to use symbolic variables in DIRECTOR commands enables the course programmers to develop generalized logic modules that are used repeatedly throughout a course of instruction.

**DIRECTOR COMMANDS**

The major functions performed by DIRECTOR, through DIRECTOR commands, include:

1. Accepting and interpreting commands.
(2) Retrieving text.
(3) Transmitting text to CRTs.
(4) Controlling position of cursor and text on CRT screen.
(5) Indirectly controlling presentation of materials on film.

ACCEPTING AND INTERPRETING COMMANDS

DIRECTOR accepts and interprets commands under a variety of conditions. It accepts commands from computer programs during course administration; it interprets commands from students' inquiries; it accepts commands from authors online at the CRT, at all times. For example, an author may play the role of student in his course, to check out the text and logic. To do so, he simply sits down at the student station and "signs on" as a student. At any time while he is "taking" the course, the author may initiate, from the CRT, either DIRECTOR or EDITOR commands without affecting the course in any way. He often does so in order to correct an error in the text, or to troubleshoot some aspect of the logic. One useful characteristic of the commands is that the author does not need to know the identifier for a text element he is looking at in the course in order to change it. For example, if the author wants to MODIFY a text element he is looking at while playing the role of student, he simply types in the MODIFY verb. DIRECTOR knows the identifier for the text it has just transmitted to the CRT, and will automatically use that identifier with the author's MODIFY command.

When DIRECTOR interprets a command, it replaces symbolic variables with the actual contents of the storage place specified by the symbol. In the case of Coursewriter-DIRECTOR operation, the symbolic variables may be any of the 31 Coursewriter counters or six buffers. This provides the indirect addressing feature which is so useful in implementing generalized IDM logic.

RETRIEVING TEXT

DIRECTOR retrieves text from online files on the basis of an element identifier. As discussed in Part 4, this identifier may be the actual file location (e.g., File D, Location 1976); or it may be a mnemonic character string, such as a word the student wants defined.

DIRECTOR also retrieves text from special computer storage locations; for example, it may access the contents of a Coursewriter counter or buffer for transmission to a terminal (7).

TRANSMITTING TEXT TO CRTS

DIRECTOR transmits the text it has retrieved to the CRT. It may also transmit text that is contained within the command for itself. For example:

(ROLL 'PLEASE TRY AGAIN')

will cause DIRECTOR to transmit the actual character string "PLEASE TRY AGAIN" to the CRT without destroying the text currently there.

DIRECTOR also can transmit course programs to the CRT. For example, the instructional programmer may want to see a portion of his course on the CRT. DIRECTOR, operating in what is called "typewriter mode," will transmit the coded program to the CRT. The programmer may then make changes, online, to the program itself. A sample portion of such a program is shown as Figure 10.
If a course is available that was written in standard Coursewriter, intended for presentation on a typewriter (standard Coursewriter III was designed to operate with typewriter terminals), DIRECTOR can operate in “typewriter mode,” thus allowing the text designed for typewriter use to be presented on the CRT. While this mode of operation does not take advantage of the capabilities of the CRT for conversational interaction, it does allow presentation of “typewriter” courses without reprogramming the material.

CONTROLLING POSITION OF CURSOR AND TEXT ON CRT SCREEN

As previously discussed, the flexibility and smoothness of the conversation that takes place on the CRT screen depends largely on the capability for manipulating the cursor and addressing portions of the screen. Each DIRECTOR command provides an argument for specifying cursor positioning. In addition, there is a command that provides direct cursor control, and a command that allows for changing portions of the screen without affecting text in the other portions.

DIRECTOR will remember where the cursor has been positioned and can also remember the positioning of blocks that have been assigned for different purposes. For example, when a student makes an inquiry, the answer is automatically positioned so it does not disturb existing text on the screen. When the student is finished reading the response to his inquiry, he presses SEND. DIRECTOR erases the text of the inquiry response, and positions the cursor back where it was before the inquiry.

CONTROLLING PRESENTATION OF MATERIALS ON FILM

Visual materials on film are displayed by the Perceptoscope. With this device, it is possible to display an individual frame or a moving (any speed) sequence of frames. The frames are addressed through cue marks in the sound track of the film. The instructions which control the Perceptoscope are transmitted through the CRT circuitry. These instructions tell which frame(s) to display and at what speed. The instructions may be prestored as part of CRT text elements (invisible to the student) or may be issued directly by computer programs. Hence visual materials on film may be tied directly to specific text presentations on CRT, or they may be integrated into the instructions by the course logic programs.
Part 7

PERSPECTIVE

MEANS VS. ENDS

The text-handling subsystem described in this report provides some of the means for achieving the goals of effective, efficient, cost-effective training with the aid of computers. The subsystem was designed to place as few constraints as possible on CAI research, instructional strategies, course content, and other aspects of the way in which the computer is used in an instructional environment. Tools such as the text-handling subsystem may be a necessary, but certainly not a sufficient, condition for achieving effective and efficient CAI.

The text-handling subsystem is designed to leave open such considerations as the kinds of information that should be presented to a student via the terminal. The "text" transmitted to the student may consist, for example, of the actual content of a lesson, an analysis of the student's progress to date, a prescription for further study, or a practical exercise. The decision as to what information to transmit to a particular student at any given point may be extremely simple or extremely complex; it may be based simply on the student's direct inquiry or response to a question, or on dozens of interrelated factors involving his performance history, his aptitudes, his objectives, his style of learning, his prior training, and so on. Clearly, these are decisions to be made by a particular instructional model.

Such tools as the text-handling subsystem, then, do not ensure effective and efficient instruction using computers. They do, however, provide a technological base upon which effective and efficient instruction can be built.

MEDIA

The text-handling subsystem described here centers around the use of a CRT terminal for communication between student and computer. In the present state-of-the-art, the CRT is one of the most versatile and dynamic media for interactive exchange of information; therefore, at Project IMPACT, much of the development work has centered around the use of this device.

However, the CRT is by no means the only medium for communication, either from computer to student or vice versa. Although the CRT presently provides the primary channels for communication, other devices are under study as well. A film projector, the Perceptoscope, operates under computer control in conjunction with the CRT. The Perceptoscope is a specialized, variable-speed 16mm movie projector. It is also capable of single-frame presentation with a fair degree of random access capability.

Typewriter devices were the first medium used for communication between computer and student. Although there are many disadvantages to a typewriter for some devices for student-to-computer communication that are under development at Project IMPACT are the Sylvania tablet and the voice recognition subsystem. These devices are described in Project IMPACT—Computer-Administered Instruction: Description of the Hardware/Software Subsystem, HumRRO Technical Report 70-22, December 1970 (2).
instructional uses, it is sometimes desirable to provide hardcopy to a student—for example, a study assignment. The IMPACT system includes teletypewriters that the student may use to get a hardcopy of the material displayed on the CRT screen. He presses a key on the CRT keyboard that causes the teletypewriter to copy the contents of the CRT screen.

The CRT display terminals described in this document are capable of displaying only alphabetic and numeric characters. Plans are underway to provide a more capable and flexible display system in the next generation IMPACT configuration. A variety of technologies have been studied, including plasma tubes and television techniques. The improved system will be capable of displaying graphic as well as alphanumeric materials. Also under development is an experimental terminal capable of displaying 3-D and color images. The text-handling software described here will then be extended to take advantage of increased display hardware capability.

**FURTHER DEVELOPMENT OF THE TEXT-HANDLING SUBSYSTEM**

The viability of the IMPACT approach to text handling is presently being evaluated through experience in using it for course and IDM development. The text-handling subsystem, and particularly DIRECTOR, is continually evolving in response to the requirements of the developing IDMs, courses of instruction, and research studies. It is anticipated that the approach and specific design of the text-handling and other subsystem products will be available for use as feasible for Army testing and implementation at appropriate schools.
LITERATURE CITED
AND
APPENDIX
LITERATURE CITED


Appendix A

SUMMARY OF TEXT-HANDLING TERMINOLOGY

Note: The terms *text*, *page*, *element*, and *block* are technical text-handling terms and do not have pedagogical implications. For example, an element or a page may or may not constitute an instructional unit. The relationship of text sets is illustrated in Figure A-1.

**Alphanumeric CRT**
A cathode ray tube device designed to display a predefined set of alphabetic, numeric, and special characters such as punctuation, but not to display graphic illustrations produced by plotting individual points on the screen. Limited graphics may be produced through arrangements of the characters on the screen.

**Argument**
The independent variable of a function.

**Block**
A feature of the Sanders 720 CRT that allows the screen to be divided into sections that are addressed separately.

**Character String**
A series of alphabetic, numeric, or special characters (e.g., "x y o" or "word").

**CRT**
Cathode Ray Tube. This device is similar to a television screen, and is used to communicate from computer to student. The CRT has an associated typewriter-like keyboard through which the student communicates with the computer.

**Cursor**
The blinking character on the CRT screen which indicates the place where the next character will be written on the screen.

**DIRECTOR**
Directing Text for Online Retrieval. A computer software supervisor that controls the retrieval of text from text files and the presentation of text to a student at a CRT. DIRECTOR usually operates under commands issued by course programs.

**Display**
Verb, to present text on the CRT screen.

**Dump text**
A FACS report that shows a series of text elements in the order in which they will appear in an instructional unit.

**Dynamic text creation**
The online formulation of text material from previously stored elements. The computer does not perform original writing in the sense that an author does.

**EDITOR**
Entry on Disc of Instructional Text for Online Retrieval. A set of commands used online at the CRT by members of the course development team to create, modify, and retrieve text elements.
Element

Basic unit of text for storage and retrieval purposes; it may contain one or many blocks.

FACS

File Activity Control System. A set of computer programs which produces offline reports used in developing text materials and in managing development activities.

IDM

Instructional Decision Model. The rules and strategies for deciding what instruction to present next to an individual student.

Image

The principal FACS report. It shows a text element exactly as it will appear on the CRT, plus information about the element.

Iterative development

The process by which text is improved by successive refinements.

Page

All the information presented on the CRT screen at one time; it may consist of one or many elements.

Roll On

Verb, to transmit an element of text to the CRT screen without disturbing the text already on the screen. 
Noun, the text element rolled onto the screen.

Sanders 720 CRT

An alphanumeric CRT device used at Project IMPACT. (For further information on the Sanders 720 see HumRRO Technical Report 70-22, December 1970.)

String

A FACS report that lists all the text elements containing a particular word, or sequence of characters.

Text

Any information presented to the student by the CAI system.

Text creation

Entry of newly created instruction into the computer text files.

Text files

Computer-readable files of text material on magnetic disk or drum or other computer storage device.

Text format

The arrangement of text material on a CRT screen.

Text handling

Creating, storing, maintaining, retrieving, organizing, or in any other way manipulating the text (content) of a CAI course during course development and course administration.

Text modification

Changes to text previously entered in the computer text files.

Text positioning

The placement on the screen of the text being presented to the student. The cursor on the CRT points to the next text character to be displayed.

Text retrieval

The act of locating specific text material (within the computer storage) and bringing it out to the student, or author.
<table>
<thead>
<tr>
<th>Distribution List</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 DIR DAIM MANSPO (FFCER)</td>
</tr>
<tr>
<td>3 CHFW DEF NUC ENG SANC LL BR</td>
</tr>
<tr>
<td>4 DIR WSC WASH D.C. 20305</td>
</tr>
<tr>
<td>5 DIR DAIM MANY HCD RESERVE AFFAIRS</td>
</tr>
<tr>
<td>6 DFC OF THE ATTY SEC OF DEF (EMR) ATTN SPEC POLICY STUDIES</td>
</tr>
<tr>
<td>7 ACOF CCH FLO COND DEF NUC SG HCD WDR AATN FFCFTY</td>
</tr>
<tr>
<td>8 NASA SCI &amp; TECH FACILITY COLLEGE PARK MD</td>
</tr>
<tr>
<td>9 CINC US EUROPEAN COMD ATTN SUPPORT PLAN B 93303</td>
</tr>
<tr>
<td>10 CINC USA PACIFIC ATTN GS CCH DAP SAN FRAN 96430</td>
</tr>
<tr>
<td>11 CG US JAPAN APD 96334 ATTN GEN G3</td>
</tr>
<tr>
<td>12 CG US FORCES SOUTHERN COMD ATTN GEN C2</td>
</tr>
<tr>
<td>13 CG US ARMY EUROPE APD 94034 ATTN DPMNS DIV</td>
</tr>
<tr>
<td>14 CG US ARMY TRANS RSRD COMD ATTN TECH LIB</td>
</tr>
<tr>
<td>15 CG US AD CCW DNT ABF APD ADOSA GOLD</td>
</tr>
<tr>
<td>16 CG 3RG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>17 CG 1ST ARM US ARMY ATTN ADPS-12</td>
</tr>
<tr>
<td>18 CG 3RD US ARMY ATTN DCSOT FT MCFHERSON</td>
</tr>
<tr>
<td>19 CG 3RD US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>20 CG 4TH CH F DEF NUC ENG SANC LL BR</td>
</tr>
<tr>
<td>21 CG 5TH US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>22 CG 5TH US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>23 CG 6TH US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>24 CG 6TH US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>25 PNS DF 50 PSYCHON PSYCH MENTES USACO DECC -T OD</td>
</tr>
<tr>
<td>26 CG FT US ATTN DGT 93 DNG ATTN DGT 201 FT DOD</td>
</tr>
<tr>
<td>27 DIR WALTER REED ARMY INTEG RES RD PTW ENG MED NUC HRY MED CTR ATTN NLPS &amp; PSYCH/IDAT DIV</td>
</tr>
<tr>
<td>28 CG US ARMY ENLISTED ENG MED CTR FT BENJ HARRISON DOD</td>
</tr>
<tr>
<td>29 TECH LTR BOX 22 USAODL EXPERIMENTATION COMD FT DOD</td>
</tr>
<tr>
<td>30 HUMAN FACTORS TEST DIV (ADHE) USAF HOSP EGIN AFB</td>
</tr>
<tr>
<td>31 CG FRANKFORT ARNRL ATTN MMAPA-442204-2 PA</td>
</tr>
<tr>
<td>32 RNS US ARMY DOD -45</td>
</tr>
<tr>
<td>33 4TH ARMY MG US ARMY TRANSPORTABLE SAN FRAN FT DOD -43D311102033111</td>
</tr>
<tr>
<td>34 PGuna ARMY SAFETY CNT FT DOD -43D311102033111</td>
</tr>
<tr>
<td>35 CG PACAFNY ANRL HDWY N J ATTN SUMA VCI</td>
</tr>
<tr>
<td>36 CG US ARMY DEF SUPPLY FT DOD</td>
</tr>
<tr>
<td>37 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>38 CG US ARMY DEPS DIP -2</td>
</tr>
<tr>
<td>39 CG US ARMY FT DEPS DIP -2</td>
</tr>
<tr>
<td>40 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>41 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>42 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>43 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>44 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>45 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>46 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>47 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>48 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>49 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
<tr>
<td>50 CG US ARMY ATTN DEPS DIP -2</td>
</tr>
</tbody>
</table>

Note: The table is truncated for brevity. The full table contains thousands of entries.