Issues in Interactive Interface Design and Evaluation

Robert J. Deans

A Thesis

in

The Department

of

Education

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Arts in Educational Technology at
Concordia University
Montreal, Quebec, Canada

April 1998

0-612-39418-2

ABSTRACT

Issues in Interactive Interface Design and Evaluation

Robert J. Deans

This work is a practical exploration of the interactive interface. The purpose is to provide a step-by-step guideline for the design and evaluation of interactive multimedia interfaces intended for limited-use training programs, delivering specific content to well-defined user groups. The discussions are intended for new or seasoned interface designers looking for a comprehensive collection of best practices, procedures, suggestions, observations and experiences to help conceive, design, develop, approve, test and evaluate interfaces for single or series-based interactive multimedia delivered on CD-ROM or diskette. These discussions are illustrated with interfaces from three different interactive training program series designed for pharmaceutical and biotechnical companies.

Acknowledgments

The author wishes to acknowledge the following contributors to this thesis:

Jean Lalonde, President of IC Axon, design partner on the X-Plore series and the Pacing Primer interface, and for permission to use the interface material that illustrates this work and the corresponding results data.

Thierry Semoff and Rubin Boiardi of IC Axon, design partners on the Backgrounder series interface.

All of the extremely creative and patient writers, artists, programmers and coordinators at IC Axon who gave the interfaces life, depth and functionality.

Table of Contents

List of Figures x
List of Tables xii

OVERVIEW **1**

Scope **1**

Brief History **2**

The Medical Context **5**
The Interfaces Used in This Discussion 5

Three Interactive Programs on Medical Pathologies **7**
The X-Plore series 7
The disease 7
The client 7
The Backgrounder series 8
The Diseases and Conditions 8
Hypertension 8
Asthma 8
Male Androgenetic Alopecia 9
Interactive Pacing Primer 9
The Disease 9
The Client 9

Interface Design: Literature Review **10**
The Definition 12
The Properties 13
The Process 14
Guidelines, Standards, Style-Guides, Checklists, Principles 15
User-Centered System Design (UCSD) 16
Prototyping 16

Interface Design Fundamentals **18**
Striking a Balance 20

10 Steps of Interface Design **21**

STEP 1: ANALYSIS AND OBJECTIVES **22**

Analysis **22**
Analysis Pertinent to the X-Plore Series 24
Analysis Pertinent to the Backgrounder Series 25
Analysis Pertinent to the Pacing Primer Series 26

Objectives **27**
Objectives of the X-Plore series: 27
Objectives of the Backgrounder series: 27
Objectives of the Pacing Primer: 27
Tip: Staying user-centered 28

STEP 2: INSTRUCTIONAL STRATEGIES 29

Strategies Specific to Each Program 31
The X-Plore Series Strategy 31
The Backgrounder Series Strategy 32
Pacing Primer Strategy 33

Converting Objectives into Instructional Strategies 35

STEP 3: THE INTERFACE METAPHOR 38
The X-Plore series metaphor 39
The Backgrounder Metaphor 41
The Pacing Primer Metaphor 42
Tip: Constructing a metaphor 43

STEP 4: INSTRUCTIONAL TOOLS 44
Varying learning methodologies for different learning biases 44

Presentation: Looking good, making sense 47

Synthesis: The practical side of content 49
Labs as Instructional Tools 49

Case Studies as Instructional Tools 52
Assessment 54
Evaluation 55
Diagnosis 55
Treatment 56
Follow-up 57
Tip: Creating a case study 58

Q & A as an Instructional Tool 59
Question Inversion 60

STEP 5: SUPPORT TOOLS 62

Administrative Tools 62
Intro Screens 63
Exit Screens 64
Bookmark 64

Resources 65
On-line resources 65
Resident resources 66
Adjunct print resources 68

Hypertext 70
The hypertext balance 70
Hypertext mechanisms 72
Click-on/Click-off Pop-ups 72
Rollover Pop-ups 72
Dedicated Definition Box 73
Scroll Box 73
Searchable 74
Tips: Defining hypertext lexicons 75
Hypertext Options 77
Audio Pronunciation 77
Graphics/Photos 78
Video 78

Hyperlinks 79

Interface Edutainment 80
Random Events 80
Hidden Screens or Menus 81

STEP 6: STRUCTURE 82
The Users 82
The Content 83
The Metaphor 83
Instructional and Support Tools 83
Tip: A quick structural snap-shot 84

Levels and Menus: depth vs. breadth 85
Levels vs. Menus 85

Sequencing vs. Branching 88
Assessing Structure 91
Time-on Estimate 91
Navigation-to-Interaction Ratios 92
Screen-to-Layer Ratio 92
Reciprocity Ratio 93

STEP 7: NAVIGATIONAL TOOLS 94

User Input Devices 97
Tip: Selecting an input device 98

Menus 101
Menu Types 101

Direct Manipulation 102

Common Tools 104
Help Tools and Navigation Instructions 104
Accessing Help and Navigation Instructions 106
Return, Enter, OK, Close and Exit 108
Forward and Back 109
QuickTime, AVI and Director Movies 110
Scrolling 111
Pop-ups or Dialogue Boxes 112
Standardized Tools 112
Icons 113
Tip: Validating icons 115
Orientation: The Content Compass 115
Confirmation 118
Visual Cues 118
Audio Cues 119
Transitions 121
Print Navigation Instructions 122

STEP 8: EVALUATION, FEEDBACK, TRACKING AND REPORTING 123

Evaluation of User Performance 123

User Feedback Mechanisms 125
Level 1 Feedback 125
Level 2 Feedback 126
Level 3 Feedback 126
Level 4 Feedback 127
Level 5 Feedback 128
Level 6 Feedback 128
Level 7 Feedback 129

Tracking Mechanisms 130

Reporting Mechanisms 131
Business Reply Card 132
1-800 Touch-Tone 133
1-800 Voice Recording 133
Auto-modem transfer: 134
Background e-mail: 135
Tracking and reporting in the X-Plore series 136
Tracking and reporting in the Backgrounder series 137
Tracking and Reporting in the Pacing Primer 138

STEP 9: APPROVAL AND EVALUATION 139

Approval of the Interface 140
Paper Interface Approval 140
Demo Approval 140
Designing a Demo 141
Testing a Demo 142
Approval Stages During Production 143
Approval of Interface Screens. 144
Approval of Text Content 145
Approval of AV Elements 145
Production schedules in the real world 147

Interface Evaluation 149
Usability Testing 149
Modeling 150
Heuristic Evaluation 150
Alpha-Testing 152
Spot Testing 152
In-House Alpha-Test: 153
Client Alpha-Test 154
Beta-Testing 155
Controlled Beta-Test 155
Field Beta-Test 156
Beta-testing in the real world. 157
Evaluation vs. Approval and Testing 158

Program Performance Tracking 159

STEP 10: RESULTS 161
The Evolution of Evaluation 161

The X-Plore Series Results 162
Key 162
Discussion 163
Qualitative Data 164

The Backgrounder Series Results 165
Key 166
Discussion 166

The Pacing Primer Series Results 167
Discussion 167

Summary 169

REFERENCES 172

APPENDICES 176

Appendix 1: An Interface Design Checklist 176

Appendix 2: Disc 'Real Estate' 177
Projecting 'real estate' requirements 178
Multi CD-ROM Series 179
Digital Versatile Disc 180

GLOSSARY 181

COPYRIGHT 182

List of Figures

Figure 1: X-Plore instructional strategy 31
Figure 2: Backgrounder instructional strategy 32
Figure 3: Pacing Primer instructional strategy 34
Figure 4: The X-Plore metaphor 40
Figure 5: The Backgrounder metaphor 41
Figure 6: The Pacing Primer metaphor 42
Figure 7: Main index of a Backgrounder interface 47
Figure 8: Identification exercise in Backgrounder lab 50
Figure 9: Demonstration in Backgrounder lab 50
Figure 10: Dynamic model in Backgrounder lab 51
Figure 11: Exercise in Backgrounder lab 52
Figure 12: Patient assessment in Pacing Primer case study 54
Figure 13: Patient evaluation in Pacing Primer case study 55
Figure 14: Diagnosis step in Pacing Primer case study 56
Figure 15: Treatment question in Pacing Primer case study 57
Figure 16: Patient follow-up in Pacing Primer case study 57
Figure 17: Q&A as an instructional tool with remedial loop in X-Plore series 60
Figure 18: Comparison of intro screens from X-Plore series and Pacing Primer 63
Figure 19: Index to Resource tools in Pacing Primer 67
Figure 20: Manual from X-Plore adjunct print material 68
Figure 21: Text option in Backgrounder series 69
Figure 22: Dedicated hypertext area in Pacing primer case study 73
Figure 23: Scroll box in Backgrounder series 73
Figure 24: Searchable lexicon in Backgrounder series 74
Figure 25: Audio pronunciation option in Backgrounder lexicon 77
Figure 26: Random event in X-Plore series 80
Figure 27: Hidden menus (bulletin board) in Pacing Primer main menu 81
Figure 28: X-Plore structure 88
Figure 29: Backgrounder structure 89

Figure 30: Pacing Primer structure 90
Figure 31: Menu from Backgrounder 102
Figure 32: Drag and drop direct manipulation in Backgrounder 103
Figure 33: X-Plore single-page navigation instructions 105
Figure 34: Pacing primer QuickTime navigation instructions 106
Figure 35: Backgrounder menu-driven navigation instructions 108
Figure 36: Standard use of 'Forward', 'Back', and 'Close' buttons 110
Figure 37: Confirmation dialogue box 112
Figure 38: Icons at 'Nursing Station' in Pacing Primer 114
Figure 39: Segment of orientation screen from Backgrounder series 117
Figure 40: Level Two feedback in the Pacing Primer 126
Figure 41: Level Three feedback in X-Plore series 127
Figure 42: Level Four feedback in the X-Plore series 127
Figure 43: Level Five feedback in the X-Plore series 128
Figure 44: Level Six feedback in the X-Plore series 128
Figure 45: Level Seven feedback in the Pacing Primer 129
Figure 46: BRC from X-Plore series 136
Figure 47: Evaluation screen from the Backgrounder series 137
Figure 48: Evaluation screen from the Pacing Primer 138

List of Tables

Table 1: Comparative variables in three interface designs 6
Table 2: Snapshot of the X-Plore series 7
Table 3: Snapshot of the Backgrounder series 8
Table 4: Snapshot of the Pacing Primer series 9
Table 5: Relationship between users and interface characteristics 24
Table 6: Comparison of instructional strategies between interfaces 30
Table 7: Conversion of objectives into instructional strategies and metaphor, X-Plore series 36
Table 8: Conversion of objectives into instructional strategies and metaphor, Backgrounder 36
Table 9: Conversion of objectives into instructional strategies and metaphor, Pacing Primer 37
Table 10: Content matrix with multiple access methodology 46
Table 11: Relationship between structure and total number of screens 86
Table 12: Comparative advantages and disadvantages of menu structures 87
Table 13: Input device timing 99
Table 14: Input device variables timing 99
Table 15: Classification of two levels of help 104
Table 16: Demonstration cycle for testing demo interface 142
Table 17: Results from X-Plore Hypertension series 162
Table 18: Results from Asthma Backgrounder series 165
Table 19: Results from Alopecia Backgrounder series 165
Table 20: Total file volume estimation 178

Overview

Scope

> *The interface is the heart and soul of any interactive tool, just as the program is the skeleton, the media is the flesh, and the content is the blood. It is within the interface design that the learning strategy brings life to the content.*
>
> R. Deans

The purpose of this thesis is to provide practical guidelines for the design and evaluation of interactive multimedia interfaces intended for limited-use training programs delivering specific content to well-defined user groups.

It is the intent of this discussion to share with new or seasoned designers best practices, procedures, suggestions, observations and experiences for conceiving, designing, approving, testing, and evaluating interfaces for interactive medical media delivered on CD-ROM or floppy diskette. These issues will be illustrated using three different interactive medical training programs whose interfaces were designed by the author.

It is not the intent of this thesis to address technical considerations such as delivery platforms, programming languages, graphic design theory or practice, page layout, or production coordination. There are endless reference sources on these subjects.

Brief History

Since 1975, the author has researched, designed, produced and evaluated training material delivered on traditional overhead, slide and video platforms. During that time, the typical self-guided training program consisted of a binder containing one or more video tapes and a modular self-guided print package with exercises and self-tests. Training programs intended for use by instructors in classroom presentation, included scripted delivery guides for instructors, overhead transparencies and/or PowerPoint presentation diskettes, pre-printed flipcharts, printed reference material, evaluation and correction guides, and handbooks for the trainees.

Although computer-based training (CBT) has been around as long as computers, it was only in the early 1990's that training material for most clients migrated over to the interactive computer-based platform. Thanks to the decrease in cost and complexity of both hardware and authoring software -- and a shift in delivery media from early 12" 'laser disks' to CD-ROM's -- acronyms such as computer assisted instruction (CAI) and CBT moved out of the confines of military and educational institutions to become commonplace in the vocabulary of the workforce at large. The pharmaceutical industry was no exception. The healthcare industry was quick to recognize that, compared to traditional classroom methods, the use of CBT reduced both student and trainer time, improved proficiency and reduced costs (Perez & Willis, 1989).

During the transition period, CAI/CBT programs were often hybrids, supplementing print material with rudimentary programs using command line interfaces. However, as the interactivity of the programs and their delivery platforms improved, both clients and users became more comfortable with the medium. Concurrently, the potential and effectiveness of interactive media as another training tool gained recognition. Soon the print material became supplemental to interactive training tools. Now, interactive programs provide print material more as a placebo than as a functional reference tool, if at all. The three interactive interfaces chosen for this thesis reflect this transition.

In 1993 the author became involved in the design, development and evaluation of an interactive CAI tool. Since then, he has designed and developed numerous interactive programs for pharmaceutical and biotechnical companies in Canada and the U.S., primarily as a freelance instructional/interface designer. The majority of this work has been done for a Montreal-based interactive media production company called IC Axon Inc. which grew to almost 50 employees in the first two years of its operation.

At IC Axon, programs are produced by project-specific teams (as opposed to an assembly-line approach to production). For these teams to function effectively, it is preferable that all members possess a basic comprehension of each other's field of expertise. There exists ample documentation on medical anatomy, physiology and pathology. The same applies to the disciplines of research, marketing and sales, graphic design and authoring for interactive computer programs. While preparing material for cross-functional pollination, the author found that extensive literature existed on the theory of interface design for large commercial interfaces. However, there was an absence of basic, practical, 'how-to' documentation on the design and evaluation of interactive interfaces for narrow-subject, limited audience applications.

Initial research on the subject revealed literature that painted interface design as an issue of enormous complexity, far beyond the ability of the average designer familiar only with traditional instructional media. Perhaps the complexity of many of the publications was appropriate for commercial software products intended for international distribution. But for narrow content (specific medical pathologies) and targeted distribution (specific audiences from 1,000 to 10,000), there was an evident need for something more accessible, digestible, and much more practical.

This is one of the contributing factors that prompted the preparation of this thesis.

The Medical Context

The Interfaces Used in This Discussion

The design and development of interfaces addressed in this discussion are illustrated with samples selected from three training initiatives. In addition to the interface design, the author has written major or minor portions of the medical content of some or all of the modules of the programs featured. The author has not executed any of the graphic work or programming. This discussion is intended to focus primarily on interactive interface design, the area in which the author is most interested and implicated.

To help illustrate key issues concerning interactive interface design and evaluation, the author has selected three interactive series that represent a wide range of instructional objectives, design principles and evaluation methodologies:

1. The X-Plore series
2. The Backgrounder series
3. The Interactive Pacing Primer series

These programs were chosen because their interfaces represent:

- three different interface designs;
- three different levels of complexity;
- three different client objectives;
- two different user groups;
- three different instructional strategies;
- three different metaphors; and
- three different methods of migration from traditional training material to the interactive platform.

	X-Plore	Backgrounder	Pacing Primer
Pathology	Hypertension	Systemic hypertension Asthma, Alopecia	Cardiac conduction disorders
Metaphor	Quiz	Classroom, book & lab	Nursing station
Complexity	1-3 diskettes per module	1 CD per module	Single CD
Objectives	Confirm knowledge base	New treat. strategies	Improve on basics
Audience	Field representatives	Field representatives	Critical care personnel
Strategy	Q and A	Present and synthesize	Case studies / learning-on-demand
Migration	Disks and printed booklet	Discs and resident text	Disc & resident resources

Table 1: Comparative variables in three interface designs

The X-Plore and Backgrounder interfaces are good examples of Johanassen's (1985) observation that interactivity evolved from hardware-driven to needs-driven. The instructional strategy of the X-Plore interface is a function (i.e. limitation) of its delivery medium (3.5" floppy diskettes). The subsequent Backgrounder interface provided users with much more flexibility and control through branching instruction and adaptive learning strategies.

Although these three programs do not cover all issues discussed in this paper, they help to illustrate many of the points addressed.

Three Interactive Programs on Medical Pathologies

Of the three forms of rhetorical models of presentation design (expository discourse, procedural instruction, and explanation), the three examples used in this discussion concern only expository discourse. Procedural instruction and explanations lie in the domain of medical school.

The X-Plore series

Subject	Hypertension
Producer	IC Axon Inc.
Client	Merck & Co.
Related products	Angiotensin II antagonist (blood pressure lowering agent)
Audience	Approx. 750 field sales representatives
Delivery media	1.44 MB floppy diskette series with adjunct booklets
Minimum platform	PC 486 33 M Hz, 8 MB, 16 colour, Windows 3.11

Table 2: Snapshot of the X-Plore series

The disease

24% of the adult U.S. population suffers from essential hypertension, making high blood pressure the most common risk factor for congestive heart failure. It predisposes those with elevated blood pressure to various heart diseases, such as left ventricular hypertrophy, heart failure and coronary heart disease. The latter may include myocardial infarction, sudden death, stroke, peripheral vascular disease and renal failure.

The client

Merck & Co. is one of the largest and most respected pharmaceutical companies in the world with a sales force of approx. 3500 in the U.S.

The Backgrounder series

Subject	Hypertension, Asthma, Male genetic alopecia
Producer	IC Axon Inc.
Client	Merck & Co.
Related product	Angiotensin II antagonist (blood pressure lowering agent)
Audience	750 field sales representatives
Media	CD-ROM's
Minimum platform	PC 486, 60 MHz, 16 MB, 256 colours, 8 bit audio, Win 95

Table 3: Snapshot of the Backgrounder series

The Diseases and Conditions

To date, three training series have employed the Backgrounder interface. They address hypertension, asthma and male androgenetic alopecia (male pattern baldness).

Hypertension
(see X-Plore series)

Asthma

Asthma is a chronic respiratory condition afflicting many people in North America. An estimated 5-10% of the population will suffer from asthma during their lifetime. Only 600 Canadians per year die of asthma, but the price associated with the disease is staggering. 1.3 million days of work are lost each year due to asthma-related disability, costing business an estimated $76 million. Recent figures indicate that asthma's annual direct cost to our healthcare system is approximately $306 million. If indirect costs are added, the total bill to Canadians is estimated at between $504 million and $648 million per year.

Male Androgenetic Alopecia

Male androgenetic alopecia (male pattern baldness) is a progressive condition that begins in young men and increases with age. Most common in Caucasians, its prevalence is around 50% beyond age 40, and 80% by the seventh decade of life. Studies have shown a correlation between this hair loss and depressed sexuality, poor self-concept and low self-esteem.

Interactive Pacing Primer

Subject	Cardiac conduction anomalies and pacing intervention
Producer	IC Axon Inc.
Client	Medtronic Inc.
Related product	Cardiac pacemakers
Audience	Approx. 10,000 critical care nurses and technicians
Media	Multi-user CD-ROM
Minimum platform	PC 586, 120 MHz, 16 MB, 256 colour, 8 bit audio, Windows 3.11/95

Table 4: Snapshot of the Pacing Primer series

The Disease

The average prevalence of pacemakers in the US is 2.6 per 1,000 with a peak at 26 per 1,000 by age 75. It is therefore no surprise that over 100,000 pacemakers are implanted in Americans every year.

The Client

Medtronic Inc. is the worlds leading medical technology company specializing in implantable and intervention therapies, and the world's leader in pacemaker systems, with 2.4 billion dollars in annual sales in 1996.

Interface Design: Literature Review

One shortcoming of many books and articles written on the issue of interface design is that they approach the subject from the perspective of major software development intended for large commercial audiences. Short, limited-use instructional tools targeted at specific users for narrow content areas are generally not addressed. This thesis is intended to help fill in that gap.

One limitation of some of the writing on Interface Design is evident in works that are either written by systems engineers (e.g. Intelligent Interface Design, Chignell & Hancock, 1988) or programmers (e.g. Programming the User Interface: Principles and Examples, Brown & Cunningham, 1989). As Heckel (1984) observes, since these writers tend to think logically instead of visually, their thinking does not lead to user-friendly design. On the other hand, interface designers should have a basic understanding of programming since the successful development of an interface relies on continuous interaction with the programming team.

Another stream of thinking is captured in Roth and Hefley's (1993) Intelligent Multimedia Presentation Systems: Research and Principles. Here, the uni-directional information delivery characteristics of the interface overshadow any discussion of its interactive, instructional or evaluation capabilities.

Understanding Interfaces: A Handbook of Human-Computer Dialogue (Lansdale & Ormerod, 1994) and Designing the User Interface: Strategies for Effective Human-Computer Interaction (Shneiderman, 1987) exemplify the majority of writing on the subject. While they do cover the many aspects of interface design from analysis through to evaluation, the focus is on structural and psychological issues. These two works are cited frequently in this thesis.

There is also a collection of guides and handbooks such as Multimedia on the PC: A Guide for Information Professionals (Desmarais, 1994) and the Multimedia Producer's Bible (Goldberg, 1996) which analyze all the elements of multimedia from market research through to distribution, but do not dedicate much discussion to the design process of interactive instructional media.

There are endless articles and books on cognition and the fundamentals of graphic design theory, and the technical side of interfaces such as screen elements, page layout, and programming. These include studies such as Type and Image: The Language of Graphic Design (Meggs, 1989), or The Design of Everyday Things (Norman, 1990) which addresses the psychology of design.

One extensive source for graphic guidelines in interactive media and web-based instruction is The Yale C/AIM Web Style Guide (Lynch & Horton, 1997). This manual has been particularly useful as one of the authors is a respected designer of medical instructional media for the Yale School of Medicine. Other online resources include the Association for Computing Machinery (ACM) which has an extensive digital library (www.acm.org), and the Journal of Interactive Media in Education (JIME) (www-jime.open.ac.uk/).

It is the intent of this thesis to address the practical issues that lie somewhere between the psychology and the technical issues of interface design, and to analyze the major components of an effective interface, as well as their sequencing. However, before proceeding, it is useful to present a short review on how other authors see the definition, purpose, properties, and process of designing an interactive interface.

The Definition

Interface: A surface forming the common boundary of two bodies or two faces; a device that bridges different systems, people, ideas technologies, etc.; the point at which two elements of a system join; the method of integrating these two elements (Webster, 1988).

There are as many definitions of the concept of interface as there are authors. For the purpose of this discussion, the concept of 'interface' will not be defined solely as a noun or verb. A more practical definition will be adopted.

> **Interface:** The medium between the user and the interactive computer program. A virtual environment based on an appropriate analogy meaningful to the user and related to the subject, equipped with functional tools used to interact with the structured content, learning activities, and evaluation mechanisms.

As defined by Norman (1986) the role of this 'interface' is to construct a bridge that provides execution and evaluation between the computer system and the objectives of the user and designer, so that either the system is closer to the user, the user is closer to the system, or preferably both.

The Properties

According to Eason (1988) any design should have four obvious properties:

- Functionality: The system does what it is designed to do.
- Usability: Users can easily "master and exploit" the system.
- User Acceptance: Users perceive the system as "facilitating goals."
- Organizational Acceptance: The system matches and facilitates organizational goals.

Interface properties have also been standardized by international norms committees. In a recent EEC directive, (90/270/EEC, 1990, article 4,5, annex 3) several software/system prerequisites require that an interface be:

- suitable for the task;
- easy to use;
- adaptable to the operator's level of knowledge or experience;
- provide feedback to users on performance; and
- display information in a format and at a pace adapted to the operators.

One desirable property common to all designs, designers, and design theorists is that the requirements for learning an interface should be minimized if interfaces are designed effectively (Larkin, 1989; Draper & Oatley, 1992; Lansdale & Ormerod, 1994). Therefore an effective interface should:

- enable users to manipulate the interface by mimicking learned and reflexive actions required to perform a task using the entity it emulates; and
- reduce the need to learn skills that are necessary to use an interface so that users can concentrate on learning the skills imbedded in the content.

These two issues will be discussed later in greater detail.

Having defined the properties of an interface, it is logical to also define the characteristics of an interface designer.

Contrary to Young (1983) who implies that interface design requires both a psychologist (to address the needs of the users) and a systems engineer (to design the interface structure), a good designer should be both, with the ability to bridge the gap and bring both skills together to focus on the needs of the users. Norman (1986), adds a third prerequisite skill, suggesting that an interface designer must also be a content expert as well. Therefore, the skills a designer – or the design team -- should bring to any interactive interface are:

- Knowledge of design (visual design, layout, programming).
- Knowledge of the users (cognitive science, communication, interaction).
- Knowledge of the subject. (Details provided by subject matter expert (SME), but the designer must understand the subject as well).

The Process

The interface is at the center of any interactive instructional tool. It defines the overall qualities of the program it facilitates. In a survey of 70 commercial software designers, Myers and Roson (1992) found that developers spent an average of 45% of the design phase of a project, and 50% of the implementation phase on the user interface.

A methodical approach to interface design may not always guarantee an effective interactive tool, but it will decrease the probability of finding yourself at the beta-test stage with a dysfunctional product.

Roth and Hefley (1993) offer fundamental guidance to the process of interface design, suggesting a four-step process in which the designer selects:

1. Content: the information necessary to support the objectives.
2. Technique: appropriate media and modalities.
3. Presentation design: how media and modalities will be used.
4. Coordination: the composition and organization of tools and material.

Lansdale and Ormerod (1994) offer three general approaches to interface design which have been employed in the three interactive interfaces modeled here:

1. The use of guidelines, standards, style-guides and checklists.
2. User-centered system design (UCSD).
3. Prototyping.

Guidelines, Standards, Style-Guides, Checklists, Principles

Guidelines are at the top of Savoie's (1994) list of methodologies employed to design interactive interfaces. They are defined as tested and proven techniques that address the more pragmatic issues behind interface design -- such as interaction style, data display, colour, layout, use of media, etc.

In this category, Savoie also lists 'Principles,' which he considers to be those tested and proven theories that address the cognitive processes influencing human-computer interaction.

Examples of this include Card, Moran and Newell's (1983) recognition-action cycle of perception - response - execution, and Norman's (1986) general theory of action, based on the analogy of building bridges between the users' concept of a goal and the actions needed to accomplish that goal, and between the results provided by the system and the users' understanding of their progress.

User-Centered System Design (UCSD)

Norman (1986) considers a design to be user-centered when:

- The design supports the information needs of users in a natural and intuitive way through the selection of the metaphor(s), and how learning strategies are converted into instructional tools within that metaphor.
- The users are involved in the design of the interface. For example, the Pacing Primer interface involved a prototype test of the interface, and subject matter experts (SMEs) that represented both the client and users.
- The technology is appropriate for the social and organizational context in which it will be used. Computers and software were common tools for the target users for all three interfaces discussed here.

Prototyping

Myers and Roson (1992) recommend prototyping tools which they classify as: window managers (e.g. visual basic); toolkits (e.g. menus, buttons, scroll bars); interface builders (e.g. Icon Author, Authorware); and User Interface Management Systems (e.g. Director). However, unless an accurate facsimile of the final interface is used in a prototype exercise, the value of the data collected is questionable. Lansdale and Ormerod (1994) express similar concern in that:

- Prototypes encourage designers to focus only on the user interface, often ignoring fundamental problems of systems analysis.
- Because of the time and resources invested, there can be a reluctance to discard the design, or to find alternatives if it proves to be unsuitable. This can result in inherently poor designs being 'patched up' for service.

The approach used in the development of the three programs reviewed in this discussion embraces the use of the best existing tools, where appropriate, and keeping a UCSD focus during the design stage. Due to the enormous costs in terms of time and financial resources, prototyping is only recommended for large scale projects or 'series' where the initial investment can be amortized over a series of programs. At the same time, the developers should be prepared to either discard or re-engineer a design that does not meet the pre-determined minimum user acceptance during the test of the prototype.

One element common to the examples presented here is that that interactive interface design began with, and remained focused on, the users in all steps through to final delivery (within parameters defined by the client). While guidelines, standards, style-guides, checklists and UCSD were employed in the development of the X-Plore and Backgrounder series, only the Interactive Pacing Primer had the luxury of using a prototype approach to interface design.

Interface Design Fundamentals

There are probably as many rules, principles, guidelines, and theories as there are designers and authors. Understandably, no single set can satisfy all design requirements. The best and most appropriate design strategy is often a combination of many separate elements orchestrated to meet the diverse needs of each program and the target users.

Lansdale and Ormerod (1994) caution that:

- Interface use is context sensitive: Design for the task and the users.
- Interface design is not simply about the ease of learning: Don't oversimplify an interface solely for the sake of ease of use.
- Technology alone should not drive design. Select technology according to interactive and learning objectives, not vice versa.
- Psychological theory can be difficult to apply to design: there are many different theories for the many different functions of an interface.

Norman (1986) is more philosophical, suggesting that designers:

- Use both knowledge in the world and knowledge in the head.
- Simplify the structure of the task.
- Make things visible: bridge the gulfs of execution and evaluation.
- Get the mapping right. Make sure that users can determine the relationships between:
  - intentions and possible actions;
  - actions and their effects;
  - the system state and what is perceived; and
  - the perceived system state and the goals and intentions of the users.
- Exploit the power of natural and artificial constraint.
- Design for error.
- When all else fails, standardize.

Others, such as Molich and Nielsen (1990), are more pragmatic, suggesting that interface designers:

- Use simple and natural language.
- Speak the users' language.
- Minimize the users' memory load.
- Be consistent.
- Provide feedback.
- Provide clearly marked exits.
- Provide shortcuts.
- Provide good error messages.[1]

To this list Ben Shneiderman (1987) adds:

- Strive for consistency.
- Enable frequent users to use shortcuts.
- Offer informative feedback.
- Design dialogs to yield closure.
- Offer simple error handling.
- Permit easy reversal of actions.
- Support internal locus of control.
- Reduce short-term memory load.

Some are quite extensive. Brown and Cunningham (1989) offer an exhaustive list of no less than 93 interface design guidelines and principles.

---

[1] This design principle (which appears on many lists) suggests planned design flaws. A good design will be operationally error free, and therefore will not require error messages.

Striking a Balance

Norman (1986)[2] recognizes fundamental tradeoffs between information, space and time. As information and objectives increase in both volume and complexity, the number of design strategies increase as well. And as the complexity of the interface design and the number of instructional tools increase, the demands on users' short and long-term memory escalate, distancing them from the information and learning experience. Eventually, the interface will reach an undesirable crossover point where its complexity will begin to work against its effectiveness. This point will be different depending on the users and the design.

Identifying and avoiding that point is part of the art and science of interface design. Designers must strike a balance between client objectives, information volume, interface complexity, users' ability, platform and delivery medium limitations, and the reality of a budget and schedule.

[2] Norman also draws an inverse relationship between informativeness and both space and system responsiveness, but this is no longer an issue with the level of technology now available.

10 Steps of Interface Design

Design principles, theories, rules and guidelines should be referred to from conception through evaluation. The designer should selectively adopt (with moderation) those most suitable and appropriate for the objectives and users. Often missing from the literature and the lists of interface design properties and procedures are the practical mechanics that link the steps into a process. This thesis will offer the basic ten step procedure that was used successfully in the three interactive instructional interfaces that will be used as examples.

I. **Analysis and Objectives:** Define global objectives based on client needs, content complexity, user needs and characteristics and the delivery environment.

II. **Strategies:** Select the appropriate learning strategies for the objectives and learner types.

III. **Metaphor:** Find an appropriate vehicle/analogy for the learning strategies.

IV. **Instructional Tools:** Populate the metaphor with instructional tools that represent the learning strategies.

V. **Supplementary Tools:** Fill in any instructional gaps and polish the interface with supplementary support tools.

VI. **Structure:** Build the structure(s) required to support the interface and its instructional and supplementary tools.

VII. **Navigational Tools:** Create the necessary navigational tools to pilot users through the structure.

VIII. **Evaluation:** Integrate the necessary feedback, evaluation, remediation tracking and reporting mechanism into the interface.

IX. **Approval and Testing**: Run the program through alpha and beta testing.

X. **Results**: Collect the data and use it to improve further initiatives.

Step 1: Analysis and Objectives

Front-end analysis is the de-facto process that precedes the research, design and development of any instructional tool. In the three cases discussed in this thesis, the intended audience was clearly defined as a result of earlier, non-interactive training initiatives. Delivery platforms and other technical parameters were specified by the client. As a result, the only additional front-end analysis conducted involved using 'focus groups' to collect basic information on the intended users' experience and attitude regarding interactive computer-based training. In general, the analysis techniques used at the 'front-end' to construct objectives for these audiences did not differ greatly from those used in the development of any other traditional linear instructional tools.

Analysis

According to Lansdale and Ormerod (1994), the task and the users should be modeled into the interface design. Analysis of users, task and delivery enables the designer to study the context in which an interface will be developed and delivered. The subsequent requirement specifications assist in choosing from a wide range of instructional and navigational tools. These tools must ultimately create an effective, efficient and interesting environment for the content, audience, client objectives and delivery platform. That environment is the interface.

It is generally accepted that a front-end analysis should consist of:

- the objectives defined by the client;
- the content volume, depth and complexity
- the physical environment and time context in which the program will be used;
- technical analysis (delivery medium, operating system and platform specs);
- the needs of the users (expectations, personal objectives); and
- users' characteristics (literacy, memory, interpretation, patience/enthusiasm, skills, prior experience, needs, etc.).

In the case of the three programs discussed in this thesis, the clients supplied clear, finite objectives based on specific content. One advantage of medical health science is that content can be clearly defined.

The platform for the X-Plore and Backgrounder series was one specific brand and model number of laptop computer used by sales representatives. The reps ran the program at their own pace (at home, at the office and while traveling) within several weeks of reception.

The environment for the Pacing Primer differed in that it involved a broader base of PC's, some in a multi-user environment. As a result, the auto-install program had to be more comprehensive, and the technical requirements more forgiving. The critical care nurses for whom it was intended also used the program at their leisure, at home or at their hospital or clinic.

The users for whom the three interfaces were designed fell into the category of 'Knowledgeable Intermittent', as defined by Shneiderman (1987) in the following table.

Characteristics of Users	Response of Interface Design
Novice:	
• no syntactic knowledge • little semantic knowledge • shallow knowledge of task • anxious about using computers	• restrict vocabulary • few choices • lots of informative feedback • simple printed manuals and resident tutorials
Knowledgeable Intermittent:	
• some syntactic knowledge • good semantic knowledge	• use recognition • frequent prompts • help menus • well organized reference manual
Frequent:	
• familiar with semantics and syntax • looking for speed and efficiency	• limited keystroke input • use of macros and shortcuts • rapid response times • brief infrequent feedback

Table 5: Relationship between users and interface characteristics

Analysis Pertinent to the X-Plore Series

An anti-hypertensive drug known as an 'ACE inhibitor' had been one of the most successful agents in the company's history, accounting for well over a billion dollars in annual revenue. As the drug approached the end of its patent life protection, a new, more specific-acting agent had been developed. It was important that the sales force understand the action of the new agent, and how to position it next to its successful predecessor when detailing the new drug to doctors. The sales force was already familiar with the subject of hypertension as a result of earlier self-paced training initiative provided when ACE inhibitors had first been introduced. Pending FDA approval of the new agent, the client wished to assess their sales force's knowledge on the subject in order to refine and target subsequent training initiatives.

Unique to this initiative:

- There would be no roll-out or launch to familiarize the sales force with the new CAI software. The programs had to be simple: load, click and run.
- The modules in the series had to be delivered on 3.5" floppy disks.
- Each module had to collect data on what the sale force's perception of doctors' prescribing habits and decisions would be once the new agent was released. This would assist in the design of future instructional courseware to assist the agents in correctly marketing and positioning the new agent.

Analysis Pertinent to the Backgrounder Series

Prior to the launch of a new agent, the client would distribute self-paced instructional binders known as 'Medical Backgrounders' to the sales force. Developed by the Medical Services department, the role of these self-guided programs was to familiarize the representatives with the history, anatomy, physiology, pathology, etiology and treatment strategies behind a disease, condition or syndrome for which a new pharmacological intervention was being prepared.

This print-based method of preparation had several shortcomings:

- This approach had been used for every single agent launched. Some representatives had shelves of such binders. The tool was becoming 'worn' and therefore loosing its effectiveness.
- The self-paced instructional binders are long and dry, and not very appealing.
- Although the binders contained frequent and numerous quizzes, they had no method of collecting data on how well the users were performing, and no way of evaluating the true effectiveness of the binders.
- The recipients of the binders were beginning to complain of 'training overload' because of the amount of time required to complete each of the self-guided training binders, compounded by the increasing number of these programs.
- The production, duplication and delivery (by courier) of close to 1,000 binders per initiative was expensive.

Additional challenges that the program had to address were:

- The Medical Services department was not comfortable with abandoning the extensive print material they had developed.
- The time available before the launch date was traditionally very brief.
- The program would be step-released in five modules on five CD's, so as not to inundate the users with one huge training package.[3]

Analysis Pertinent to the Pacing Primer Series

Field representatives found they were spending too much time in hospitals and clinics (their core business group) providing training on electrophysiology and pacing basics. This detracted from the time they should have been devoting to more critical issues associated with their products and cardiac pacing.

This problem was also apparent at the 1-800 technical help line which was logging a large number of frequently-asked-question (FAQs) that could be attributed to an absence of basic knowledge regarding the fundamentals of electrophysiology, cardiac conduction disorders, and pacing intervention.

Additional challenges that the program would have to address in meeting the objectives were:

- There would be no adjunct print material or resident print manual(s).
- The program duration could not exceed three hours due to the maximum number of continuing education (CE) credits that could be offered.
- There was no standard PC platform configuration or level of Windows operating system.
- The program had to support multi-users.

---

[3] Originally the modules were to be released with sufficient time in between for feedback from earlier modules to be used in improving the design of subsequent modules. Unfortunately, the schedule was 'collapsed' to a point where this could not be done.

Objectives

Objectives of the X-Plore series:

- Measure the sales force's general knowledge level of the subject so that the client could determine what depth of training would be necessary.
- Make the sales force aware of their own strengths and weaknesses in fundamental issues regarding the subjects.
- Provide the sales force with a brief 'refresher' overview of the subject, guaranteeing a minimum knowledge level on basic and critical points.
- Help the sales force make a painless transition from traditional self-paced print-based 'binder' training to intro-level CAI.

Objectives of the Backgrounder series:

- Introduce a training tool that capitalized on the response of the previous X-Plore series, migrating to a more comprehensive training platform.
- Introduce a new and more effective training methodology.
- Ensure that users were exposed to more in-depth, requisite content.
- Decrease users' training time compared to traditional tools.
- Provide users with the opportunity to 'apply' the new material in context.

Objectives of the Pacing Primer:

- Provide a comprehensive general pacing primer that would cover all areas of basic electrophysiology and cardiac pacing.
- Provide training that would adjust to the varying needs and entry levels of approximately 10,000 nurses and technicians working in electrophysiology.
- Ensure a measurable level of competence which would enable users to apply for continuing education (CE) credits from a professional association.
- Provide a 'user-friendly' training tool for an audience with varied computer platforms and varying computer skills.
- Provide useful training tools that would encourage users to continue to access the disk as a professional reference.
- Provide a training and reference product with sufficient appeal that would encourage users to circulate the program among peers, expanding its distribution base beyond 1:1 (one disk per person).

Tip: Staying user-centered

It is not unusual to become design-centered as the potential of the interface evolves, and budget/time-line centered as the delivery date approaches. To avoid this subtle and sometimes unnoticeable shift, make a list of users' needs and the client's objectives. Keep a copy of this list on the front cover of all production documentation folders and binders. In addition, keep the list pinned up on the walls of the design team and the production department during the entire production period. This promotes on-going application of the principles of user-centered design.

Step 2: Instructional Strategies

> *"...the fundamental cognitive capabilities of users remain the same, and as a consequence, the issues that affect users' interactions with these interfaces remain largely the same."*
>
> Lansdale and Ormerod (1994)

Studies by Green and Gilhooly (1990), confirm that learning strategies should differ by the type of learner. Fast learners use mapping and exploratory procedures and slow learners use trial and error and repetitive procedures. Although sales representatives generally have high educational levels and are considered 'fast' learners, a stepped strategy that adjusted to users' familiarity with interactive training tools was used for the Hypertension initiative across the X-Plore and Backgrounder series. The introductory X-Plore series employed a more fundamental interface based on trial and error and repetition. Once feedback from the X-Plore series confirmed that users' were comfortable with self-guided interactive instruction, the follow-up Backgrounder series moved to more complex and comprehensive strategies employing mapping and exploratory exercises. This was in direct response to users' growing interest level and abilities measured by the earlier X-Plore series.

The development of the learning strategies for the Pacing Primer program was aided by the beta-test of a prototype of the interface. This identified the learning strategies to which the target audience did -- and did not -- respond.

Lansdale and Ormerod (1994) suggest that interface users rely on learning skills from three sources:

- Rote-learned procedures: Key information 'chunks' presented in objectives, overviews, presentations, reviews, quizzes, exercises, labs, case studies and exams. In the Backgrounder series, redundancy levels were as high as seven.
- Conceptual models: Conceptual models such as labs and patient case studies maintain a high interest level and are effective instructional strategies because they promote content synthesis.
- The interface: The interface promotes learning because it is built around a metaphor, which is a conceptual model itself.

These strategies were designed into the interfaces as follows:

X-Plore	• trial and error strategy • rote learning procedure • instructional interface
Backgrounder and Pacing Primer	• exploratory strategies • conceptual models • alternative problem-solving methods • effective/flexible use of examples

Table 6: Comparison of instructional strategies between interfaces

The instructional strategies common to all three interfaces used as examples in this thesis reflect Gagné's (1985) nine events of instruction:

1. Gain attention (novelty of medium, engaging interface)
2. Provide objectives (through 'Objectives' or 'Overviews')
3. Stimulate recall of prior learning (attach short term to long-term memory)
4. Present distinct stimuli (AV presentation of content)
5. Guide learning (content organized in natural learning hierarchy with appropriate advanced organizers and review components)
6. Elicit performance (labs and other activities that promote synthesis)
7. Provide feedback (correction of labs, case studies, quizzes, etc.)
8. Assess performance (tracking mechanisms)
9. Enhance retention and transfer (redundancy, synthesis and simulation)

Strategies Specific to Each Program

Fundamental instructional strategies (advanced organizers, content delivery, review, application) are integrated into all elements of an interactive design. The following program-specific strategies demonstrate how objectives extracted from front-end analysis can be aligned with appropriate instructional strategies.

The X-Plore Series Strategy

A key objective was the evaluation of the sales force's existing knowledge base, while a key limitation was the delivery medium. A Q & A approach was selected because it provided the necessary interactivity and measurement, yet could still fit onto one or two diskettes per module. It was also necessary to measure users' responses to the interactive platform as a training delivery tool.

Although the X-Plore series was primarily an evaluation tool, an instructional component was added to increase both effectiveness and interest. This was achieved by incorporating content into the interface itself via a series of quizzes using a six-step content delivery and confirmation format:

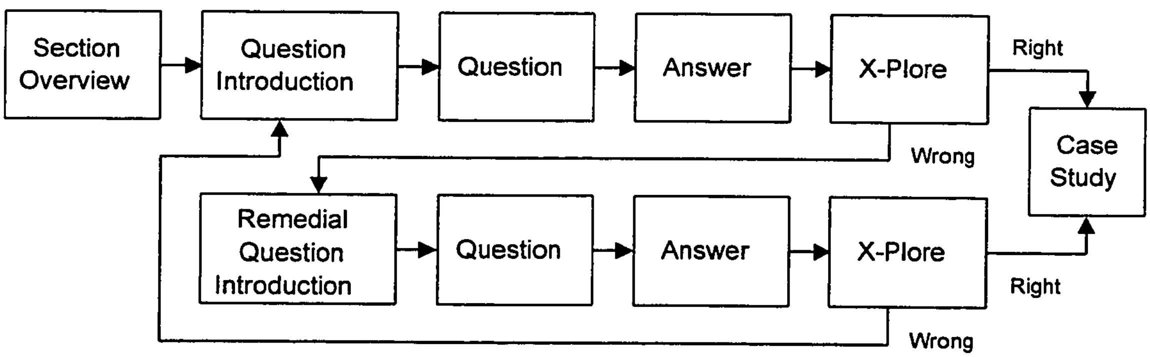


Figure 1: X-Plore instructional strategy

The Backgrounder Series Strategy

'Backgrounders' are self-guided, print-based, study programs that familiarize sales representatives with the underlying pathology, related physiological mechanisms, and pro's and con's of accepted treatment strategies.

The client wished to maintain the instructional image of the original print-programs, so a traditional classroom analogy was chosen. The basic learning methodology involved a three-step approach of content presentations, synthesis (through interactive exercises in a 'lab'), and a final exam.

Users began by selecting and reviewing specific 'lessons' from a 'workbook' interface, then moving to a 'lab' area for practical application of the recently acquired content. Labs provided an opportunity to combine new material with existing knowledge, improving transfer from short term to long term memory. Once users had reviewed all of the module's content, and completed all of the labs, they could proceed into an 'exam room' to write a 'final exam'.

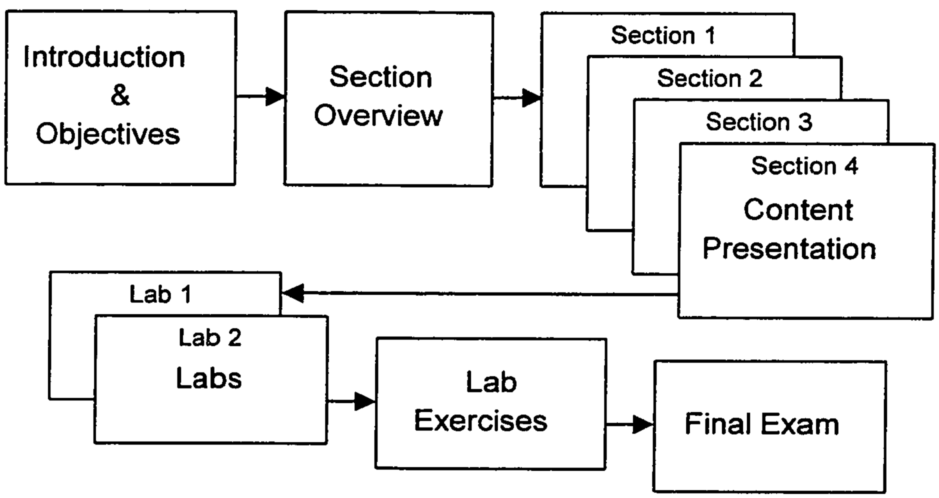


Figure 2: Backgrounder instructional strategy

Pacing Primer Strategy

The target audience for the Pacing Primer was practicing critical care nurses. Providing them with a learning environment that best approximated their own working environment was an obvious strategy. Each day nurses are faced with patients presenting a variety of problems for assessment. This lent itself to a case study approach in which users followed several patients -- independently or concurrently -- in a clinical environment, from admission through assessment, diagnosis and treatment. A parallel interface provided the necessary synthesis of existing and acquired knowledge in a low-risk clinical environment.

Because of the broad range of education and experience among nurses, a 'learning-on-demand' approach was also favored. This enabled users to select the required depth of learning, and did not force knowledgeable users through content they had long since mastered. At the same time, it provided the necessary training for those that required more skill development than others.

A traditional approach to adjusting content to user requisite knowledge is to process all users through a pre-test, then -- based on their results -- direct them to related learning activities that fill in the gaps. However, for many this would be their first exposure to CAI. Opening the program with a traditional pre-test was not attractive.

The alternate approach was to go directly into the case-study instructional vehicle. As users progressed through each case, they could access numerous resources selected according to their individual requirements (inversely proportional to their existing knowledge base). The instructional components of the program were accessed only when needed, only to the depth needed, and only by those who needed them.

This learn-on-demand design was extended through to the post-test. Users could access the post-test at any time, even immediately upon entering the program. However, it was recommended that they write the test only after reviewing the resource material and completing the case studies. If users had the requisite knowledge, skills and confidence to demonstrate mastery by successfully completing the test (80%), they were not forced to go needlessly through resource material or case study exercises.

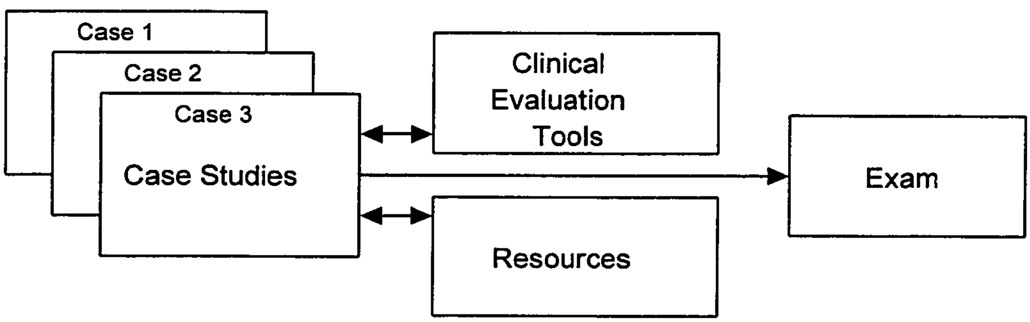


Figure 3: Pacing Primer instructional strategy

Converting Objectives into Instructional Strategies

The design approach used in the interfaces discussed here is in line with what Whiteside, Bennett and Holtzblatt (1988) refer to as 'usability engineering' where an interface is designed to meet specific objectives. To ensure that objectives are being converted into appropriate learning strategies, all components of the interface must be directly linked to client requirements. The same requirements also serve as a reference for the evaluation methodology as well.

The following tables illustrate how program objectives were aligned with learning strategies, then converted into instructional tools within the interface analogies.

The X-Plore series

Objective	Strategy	Interface
Measure sales forces' general knowledge level of specific pathology	• multiple choice testing • performance scores returned via BRC	Multiple choice tests
Ensure minimum content knowledge level	Brief overviews and summaries preceding and following each test section	Multiple choice test
Painless introduction to CAI	• minimum number of decisions and actions • chop binder to quick reference booklet	Linear, simple, consistent screen design, few branches and decision
Distribute on floppy diskettes	Simple brief information	Text-based screens, min. graphics, no sound
Feedback on sales force reaction to CAI training delivery platform	Evaluation questions	BRC cards: • quantitative scores • qualitative feedback

Table 7: Conversion of objectives into instructional strategies and metaphor, X-Plore series

The Backgrounder series

Objective	Strategy	Interface
Measurable results	• mandatory final exam • all results returned by code	Exam room
Migrate users from traditional print-base to CAI	Same structure and labeling as binders: modules, sections, self evaluation quiz, etc.	• classroom, • reference binder contains text
Expose users to all of the requisite material	All material must be reviewed to write exam	'Map' to track progress within interface
Opportunity to 'apply' the new material	Synthesis of new material w/exist. knowledge	Lab-type environment with 'dynamic models'
Decrease training time	Brief, narrated A/V type presentations capitalizing on graphics, animation and video	AV 'lectures' in 'animated talking book' environment
Maintain 'binder' text	Access to resident binder documentation	'Text' button provides portion of traditional binder text relative to that section

Table 8: Conversion of objectives into instructional strategies and metaphor, Backgrounder

The Pacing Primer

Objective	Strategy	Interface
Comprehensive general pacing primer	• resources addressing key content • skill synthesis (practical application)	• reference tools • patient case studies
Training adjusts to varying needs and wide range of entry levels	Learning on-demand	Reference 'book' resource section
Ensure a measurable level of competence	• post test • reward for passing test	• a test • professional CE credits
Provide a 'user-friendly' training tool	familiar environment with familiar tools	Nursing station with typical tools
Encourage users to continue to use the discs	• provide useful resource tools • ECG-based case studies	Comprehensive reference book with single screen lessons
Encourage users to circulate disks for exposure	• provide meaningful 'real-life' simulation • easy to navigate	• assess and diagnose patients at clinic • icon desk tools

Table 9: Conversion of objectives into instructional strategies and metaphor, Pacing Primer

Step 3: The Interface Metaphor

> *"...interface metaphors should be simple, familiar and logical to the audience.... The best information designs are the ones most users never notice."*
>
> Lynch and Horton, 1997

One fascinating element associated with an interactive tool is the interface analogy or metaphor. The interface metaphor is the 'key' to any interactive design. It is the soul of the program, giving the interface life and character, quality and attributes. From it stems all instructional activities, evaluation methodologies and navigational tools. If there is any 'art' in the science of interface design, this is where it is exercised.

The job of the metaphor is to create a familiar, functional and attractive environment where the interface can capitalize on users' pre-existing knowledge. It enables users to infer, without prompting, what actions they can take, and how the system will react (Savoie, 1994). Users quickly becomes comfortable with the interface, with a minimum of effort. This is particularly important for interfaces designed for limited-use and 'narrow' content.

Laurel (1986) refers to the metaphor as 'memesis' (sic), with transparent, closed, knowable, and consistent characteristics. Unobtrusive to the navigation and learning process, a good interface is transparent to users, yet contributes to an overall mood. Laurel underscores consistency as crucial when using semantics, icons, symbols, tool placement, and in action/reaction. It is equally important that these components consistently reflect the users' reality as well.

The metaphor provides what Lansdale and Ormerod (1994) refer to as a fundamental model or simplified representation stripped of complexity. When selecting a metaphor, it can sacrifice accuracy for the benefit of effectiveness providing it maintains a representation of reality. The designer must also ensure that the interface metaphor permeates all screens of the program, and not just the main menus. This is a common fault of many interface designs in both disk/disc and web-based interactive environments where an analogy is well constructed for the 'splash page' and main menus, but it falls apart -- or is inappropriate or altogether absent -- in following screens and levels.

The X-Plore series metaphor

Two elements that contributed to the X-Plore interface metaphor were the limitation of the delivery medium and the key objective of performance measurement. This required modifying the Q&A approach to accommodate content delivery, measurement and remedial evaluation -- all on 1.4M diskettes. The analogy chosen for the X-Plore series was simple. Since the program was in effect a pre-test, the quiz function was the obvious metaphor. The inclusion of content gave the interface more substance, above and beyond 'just another test'.

This also addressed several other objectives and restrictions:

- Mostly text-based Q&A helped limit each module volume to one or two diskettes.
- The quiz format provided a simple, familiar interface with limited navigation, and minimal navigation instructions.
- Q&A was a familiar vehicle, easily ported to the interactive platform.
- Familiarity facilitated confidence in navigating a self-guided CBT environment.
- The quiz format provided users with feedback, and the training department with quantitative performance information.

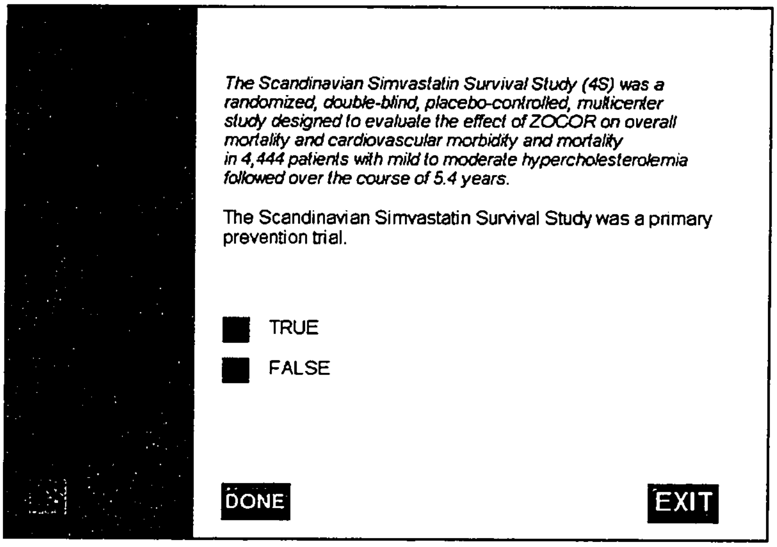


Figure 4: The X-Plore metaphor

The Backgrounder Metaphor

Backgrounders were originally delivered in a self-guided, print-based 'binder' format. For the pilot Backgrounder CD-ROM series, the classroom was a natural analogy because the exercise was one of fundamental education in the basics of hypertension. The metaphor enabled the migration of self-guided instruction out of the binder and into a simulated classroom, while keeping the basic learning methodologies with which the user-base was comfortable.

The classroom metaphor was simple, but comprehensive because of complementary sub-metaphors such as:

- formal presentations or 'lectures' (classroom, blackboard, books);
- opportunity to practice/synthesize newly acquired content (labs);
- supplementary resources (dictionary, text);
- user-tracking of navigational and work completed (globe/map); and
- a mandatory test (final exam room).

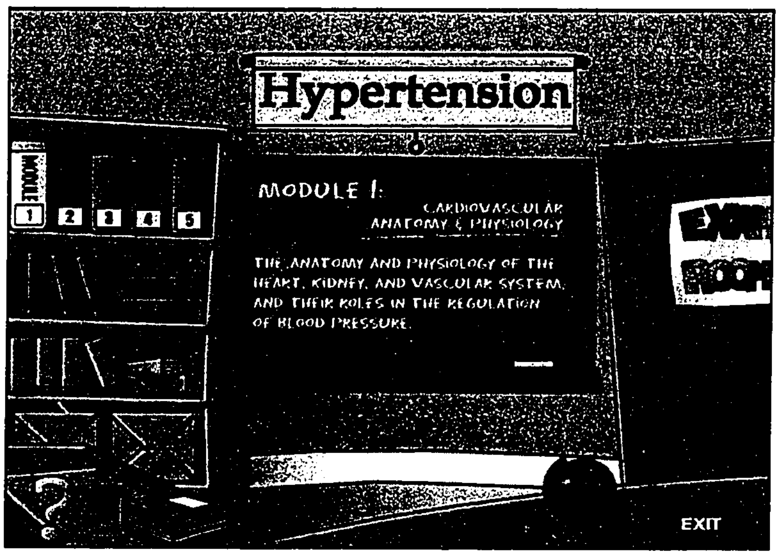


Figure 5: The Backgrounder metaphor

The Pacing Primer Metaphor

The Pacing Primer was designed for healthcare professionals working in electrophysiology. The nursing station metaphor was chosen because it is an environment common to all those working in a hospital or clinic. A visit to a cardiology ward revealed the nursing station to be the heart of the operation from which critical care staff had access to an array of tools and technology.[4] Populating the interface with metaphors for these tools provided supplementary 'info-on-demand' alternative learning strategies, and some navigational tools.

The nursing station metaphor addressed several objectives:

- It placed users in an environment that was familiar.
- The instructional strategy implicated numerous tools. The nursing station was the natural habitat for them.
- The tools served a dual-purpose, providing instructional strategies while operating as navigational icons leading to instructional or reference tools.

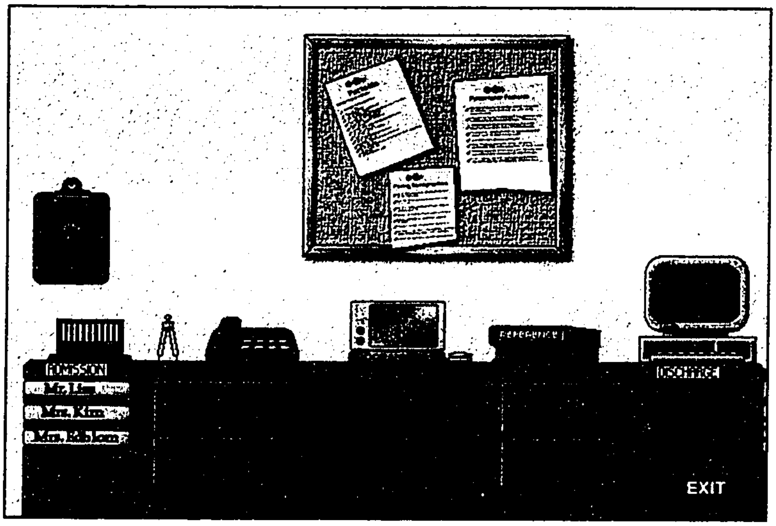


Figure 6: The Pacing Primer metaphor

[4] This video was also useful to the graphic design team who frequently referred to it for visual ideas or accuracy.

Tip: Constructing a metaphor

The following procedures are practical guidelines to creating interface analogies:

- Research and collect available print and AV material on the client and/or subject.
- Review and evaluate a cross section of different instructional methods used in existing instructional material on the same or similar subjects, either by the client, their competitors, or independent sources.
- Visit the client's web site (and their competitors'), and follow any suggested links.
- The web is also a great place for metaphor ideas. To focus your research:
  - scan a few issues (or back issues) of popular 'net' magazines;
  - check the numerous 'Top XX Web Sites' for those praised for their interface; and
  - visit the sites, making notes of the fundamentals that work, those that don't, and why.
- Wherever possible, document the immediate work environment of the target users with video or colour stills. This can provide:
  - ideas for analogies;
  - objects for conversion to symbols/icons for instructional/navigational tools; and
  - sample imagery for screen design and layout.

Step 4: Instructional Tools

The instructional components of an interface must serve the learning strategies and accommodate the content. Although addressed separately in this thesis, there is a symbiotic relationship between the interface metaphor and all other interactive tools. While the interface is the home for all tools, the individual tools themselves must still support the metaphor of the interface as logical extensions. Instructional tools are 'mission critical' because they carry the instructional strategies delivered through the interface.

For every learning theory, there are numerous instructional strategies. The key is finding one or more appropriate instructional tools that fit well within the interface metaphor, logically support the instructional strategies, strike an appealing balance between delivering/presenting information, and encourage application/synthesis of that information.

Instructional tools such as presentations, exercises, labs and evaluation mechanisms must take into consideration the learning biases of the target users, the nature of the content, and the limitations of the delivery platform.

Varying learning methodologies for different learning biases

The fundamental instructional methodologies employed in the three medical interfaces follow Goodman's (1993) three-step approach to training for typical interactive courseware:

- Demonstrate: Presentations using appropriate AV resources.
- Provide Guided practice: Synthesis through exercises such as labs and case studies.
- Test: Quizzes, tests, exams designed to reinforce and evaluate knowledge acquisition.

One of the elements that sets interactivity apart from traditional instructional methods is that it enables different users to employ different learning strategies that best suit their learning styles. This should be reflected in the choice of instructional tools. These methods are executed through three levels of interactivity, as defined by Floyd (1982) and Hon (1982):

- Level One: which is based on defined structure and advanced organizers which are either directly activated by user choice, or responsive to what the users select, recall or perform according to content exposure.
- Level Two: where subject areas are explored without program intervention, thus enabling learning and experience through the learner's own preference, biases and methodologies.
- Level Three: where "interactivity is creativity" and program flexibility enables learning methodologies not necessarily included in the core program design.

The X-Plore series is an example of Level One interactivity. Both the Backgrounder series and Pacing Primer exploit Level Two interactivity, while the Backgrounder and Pacing Primer design provide varying degrees of Level Three interactivity through the labs, exercises. resources and case Studies.

Presentations, labs, exercises and case studies encourage uses to access content from a general information matrix.

	Module 1	Module 2	Module 3	Module 4	
Case 1	Content 1	Content 5	Content 9	Content 13	**Solution 1**
Lab 2	Content 2	Content 6	Content 10	Content 14	**Solution 2**
Case 3	Content 3	Content 7	Content 11	Content 15	**Solution 3**
Lab 4	Content 4	Content 8	Content 12	Content 16	**Solution 4**
	Quiz 1	**Quiz 2**	**Quiz 3**	**Quiz 4**	

Table 10: Content matrix with multiple access methodology

The matrix can be accessed using three different approaches:

1. Linear through traditional presentations that step through the content in a guided fashion and where the endpoint of each module is a quiz. Users are guided through the content. (e.g. beginning with 1, ending with 16.)
2. Random/linear where users select smaller linear sequences where the endpoint is a quiz. Users select modules at random.
3. Non-linear through an exploratory experience such as a lab or case study. To explore and/or solve the lab/case may require random access. (e.g. 1,6,7,11,16)

The result is a common content matrix in the form of a random access database, accessible through three different search protocols that represent three different learning strategies to serve three different learning styles. Any remedial activities should also reflect the learning strategy selected by the users.

Lynch and Horton (1997) observe that the average user may not recognize these interrelationships between the different categories of information. However, the designer should -- even if these relationships are subtly imbedded in the interface and transparent to the users.

The author hopes to explore Level Three interactivity further in future interfaces.

Presentation: Looking good, making sense

The mainstay of content delivery is a presentation in some form, the AV equivalent of an instructor.[5] The Backgrounder series is an example of content presentation delivered within a standard structure which includes:

- Introduction: Identification of the main content.
- Objectives: Statement of purpose and exit level skills that will be achieved.
- Overview: Identification of specific content within all modules or sections.
- Presentation: Main content delivery.
- Summary: Review of key learning points within the entire program. More in-depth than the introduction, but more concentrated than the overview.

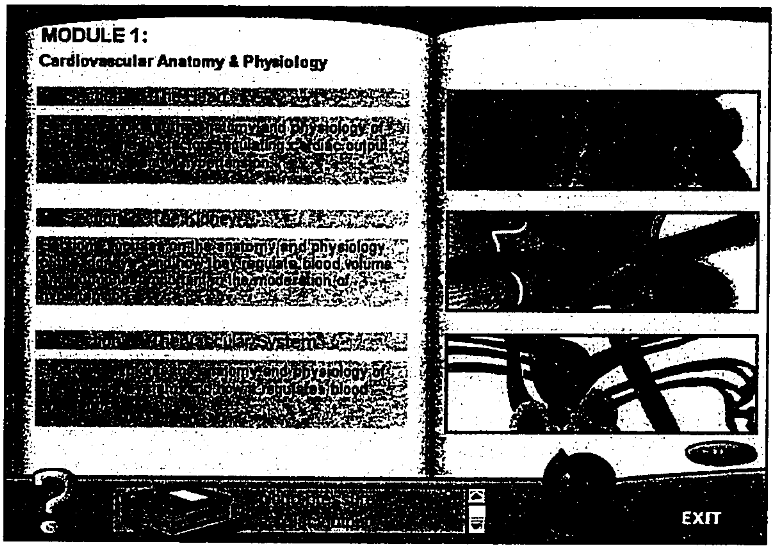


Figure 7: Main index of a Backgrounder interface

Although these components of a presentation should be viewed in sequence, users have the option of selecting them as they best see fit, given their individual requisite knowledge, and learning preferences. For example, users familiar with the subject may elect to skip the intro, objectives, and overview and proceed

[5] As demonstrated by the X-Plore series and Pacing Primer, the classic AV presentation is not always the only vehicle suitable for content delivery.

directly to specific areas of the content presentation. Users may also skip the main presentation altogether and review the overview and summary to compare the general content to their existing knowledge.

When delivering content via AV presentations (aside from the usual rules and regulations that apply to the preparation of any AV material) presentations should be designed keeping in mind several fundamentals unique to the limitations of an interface.

- Keep content light, targeting only five to seven key points per block (Miller, 1956). Retention is inversely proportional to density.
- Keep sequences short. Retention is inversely proportional to duration.
- Construct sequences in a context and hierarchy proven effective for the users.
- Provide meaningful interaction every three to five screens (Carliner, 1990).
- Keep on-screen text to a minimum. Reading is not an interactive activity, and is 25% slower on a CRT screen (Carliner,1990).
- Keep video brief and use appropriately. Watching a long video is not an interactive activity.
- Video shot for standard television is not always appropriate for, or useable on a computer screen. Shoot video keeping in mind the capabilities of the interface, microprocessor, compression codecs and graphic card.
- Most computers have a sound card of some sort. Exploit it appropriately.
- A CD-ROM (660 M) is smaller than you think. Keep interfaces simple.
- Respect CRT visual layout laws of colour, font, size, spacing, density, etc.[6]
- Keep screens clear and uncluttered. Can you read it on a 9" LCD in daylight?
- Exploit mnemonic devices and redundancy whenever and wherever possible.
- Don't forget that your objective is to help people remember.
- Research shows that people remember 20% of what they hear, 40% of what they see and hear, and 75% of what they see, hear and do (Oz & White, 1993). Exploit graphics, photos, video, narration and interaction.

---

[6] One standard is Patrick Lynch's <u>Web Page Style Manual</u> (http://www.yale.med.edu/lynch.style). Although this extensive work is targeted for the web, most of its standards also apply to CAI.

Synthesis: The practical side of content

Interactive media is effective in providing learners the opportunity to synthesize content through practical application. Researchers such as Kamouri, Kamouri and Smith (1986) have demonstrated that exploration-based training (active learning) is more advantageous than procedural-based training (passive learning). Labs and case studies are tools which promote a higher level of learning via exploration and/or synthesis of new information with existing knowledge stored in long-term memory. An evaluation component can also be integrated into synthesis-type exercises. This measurement is usually transparent to the users because it is not in the obvious form of a test or quiz.

Labs as Instructional Tools

Labs are flexible instructional tools with a structure that accommodates varied experimental application of content. Lab structures are convenient when a specific instructional tool (e.g. case study, dynamic model, patient simulator) cannot accommodate all the necessary content or instructional strategies.

The tools used in the Backgrounder series' labs can be classified in order of complexity:

Identification exercises for exploring the correct nomenclature of components of a system, locating their correct position within a system, and understanding their physical relationship to each other. This type of lab usually involves a diagram with an associated label list.

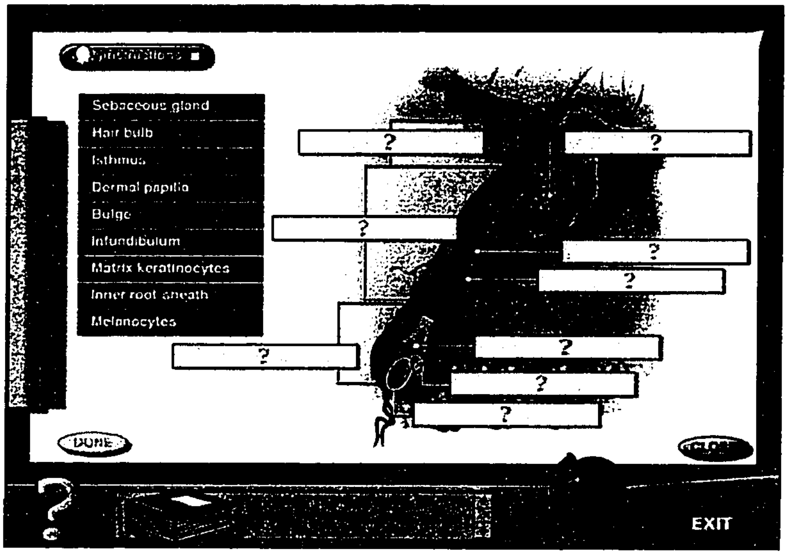


Figure 8: Identification exercise in Backgrounder lab

- Demonstrations of a system from the perspectives of its different components and the role each one plays within the system. Users select a component, then observe the sequence of events and related to its operation within the system.

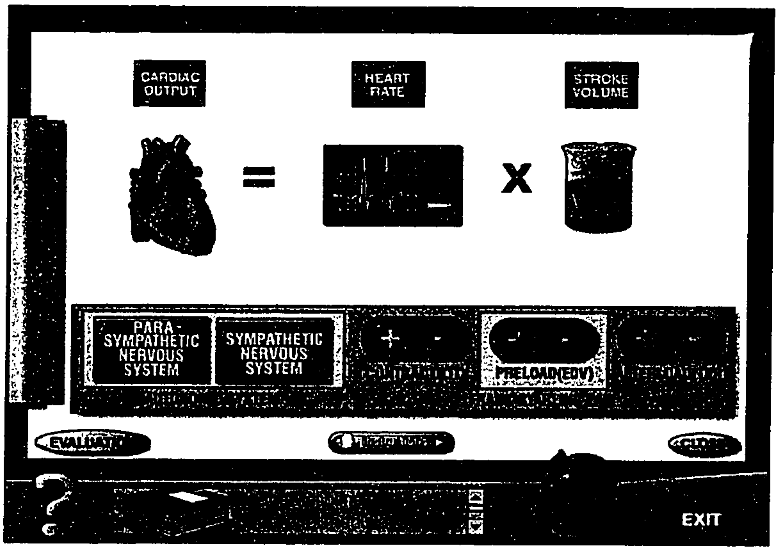


Figure 9: Demonstration in Backgrounder lab

Dynamic models to study the process, and observe how variation in different components of a process affect the dynamics of the system. Following basic instructions, users manipulate the operation of one or more components, observing the net result of their action on the entire system.

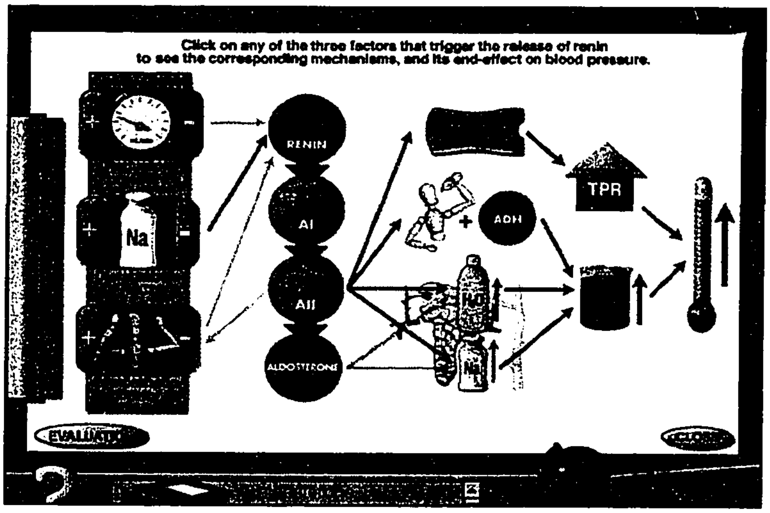


Figure 10: Dynamic model in Backgrounder lab

- Exercises in which users test their knowledge acquired during the content presentation, and reviewed and reinforced by activities during the lab. The results of the lab exercises are available to users, but do not contribute to the score in their final exam. Based on their results, users can conduct their own remedial activities by selecting and reviewing the related presentations.

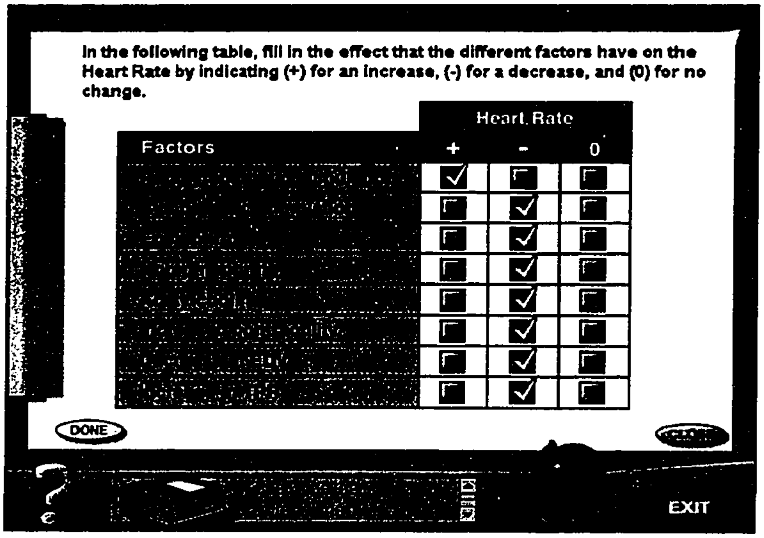


Figure 11: Exercise in Backgrounder lab

Case Studies as Instructional Tools

Case studies have long been the mainstay of the synthesis process, offering users the opportunity to apply while they learn without any negative consequences -- a definite advantage in the clinical environment. The case study approach improves critical thinking, problem solving, decision making, the ability to take responsibility, and to separate facts from inferences (Info-Line, 1987). Kamouri, Kamouri and Smith (1986) promote a computer-simulated environment because it removes any negative consequences from exploration-based training. More specifically, Oz and White (1994) observe that multimedia technology enabled medical personnel to deal with real-life clinical situations and decisions without risking the lives of their patients. Given that the Pacing Primer was targeted at critical care nurses, the case studies could not be more appropriate.

With the use of actual patient data from a variety of sources, the clinical setting and procedures can best be simulated. Aside from their popularity with users, case studies also offer designers an excellent return on investment. Once the initial interface for interactive case studies has been constructed, content can be easily changed. The cost of the program can then be amortized over subsequent case studies released as follow-up programs or refresher courses developed at a fraction of the cost. Case studies also include an evaluation component because users are asked to make clinical decisions based on medical evidence at numerous points in the progression of a disease. When constructing case studies for medical application, the rationale behind a decision is often as important as a correct or incorrect answer.

The five steps of a case study are:

1. Assessment
2. Evaluation
3. Diagnosis
4. Treatment
5. Follow-up

Assessment

During the presentation portion of an interactive program, users collect information and data. The process is similar to patient assessment in a case study, except that users control their exposure to content through the tools they select. Aside from labwork and a physical work-up, it is a well accepted axiom that 86% of diagnosis depends on what the patient says during an interview (Hampton et al, 1975). Therefore, the Pacing Primer interface included a screen where patients talked about their condition and associated symptoms on-camera in response to selected questions.

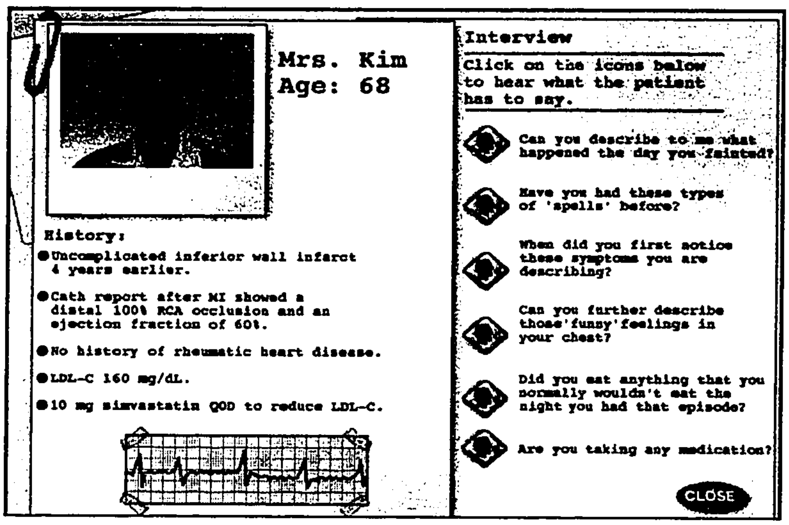


Figure 12: Patient assessment in Pacing Primer case study

Evaluation

During evaluation, users have the opportunity to formulate clinical conclusions through the synthesis of all available data collected from the assessment stage. Depending on the nature of the patient's pathology or medical anomalies, these conclusions vary in number and depth.

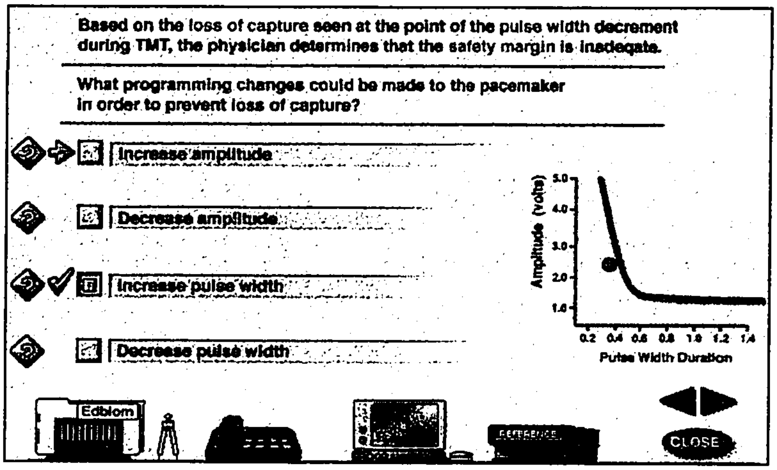


Figure 13: Patient evaluation in Pacing Primer case study

Diagnosis

Diagnosis is a procedure of synthesis in which all clinical conclusions assist in identifying a specific pathology or syndrome. Normally, diagnosis is the responsibility of doctors or specialists on staff. Case studies offer nurses and technicians the opportunity to formulate and verify their own diagnosis based on their earlier assessment and evaluation procedures.

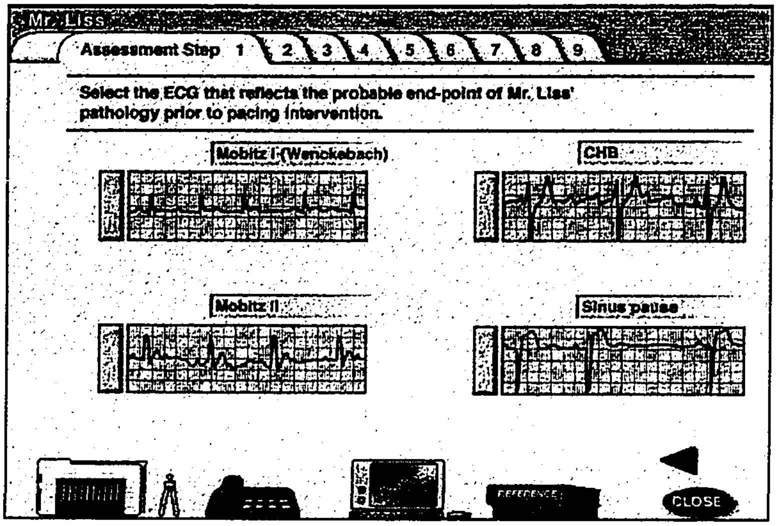


Figure 14: Diagnosis step in Pacing Primer case study

Treatment

Although the mechanics of treatment are usually executed by nurses and technicians, the selection of a treatment strategy is normally the exclusive domain of doctors or specialists. Treatment is another form of synthesis where the properties of a diagnosis are matched with medical evidence from various proven treatment strategies, usually the result of large scale studies. Here again, case studies offer nurses, technicians and even field representatives an opportunity to select a treatment strategy based on their knowledge of appropriate and successful medical intervention.

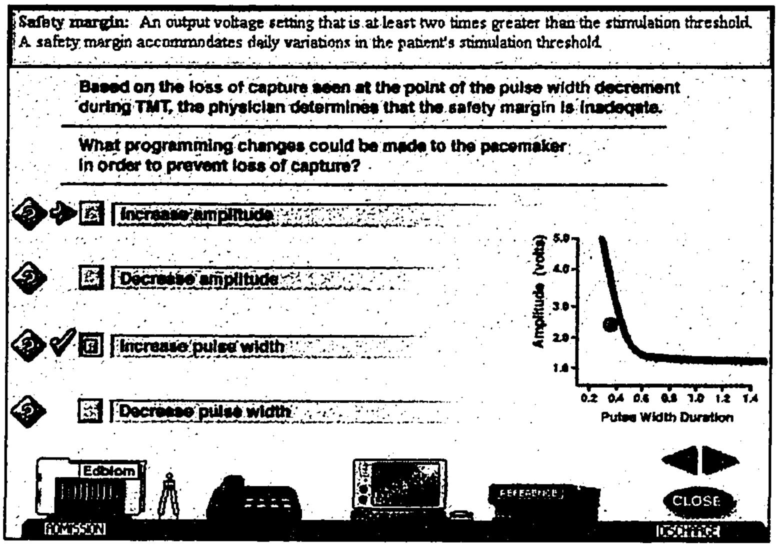


Figure 15: Treatment question in Pacing Primer case study

Follow-up

Follow-up is a standard clinical function that evaluates the success of the treatment course. In the case studies, it also provides closure to each case, which in real time could happen weeks, months or years later.

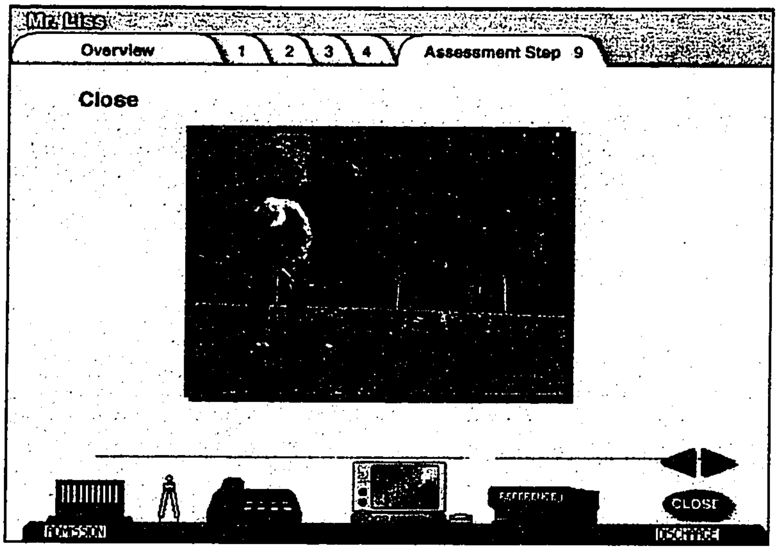


Figure 16: Patient follow-up in Pacing Primer case study

Tip: Creating a case study

Info-Line (1987) outlines four elements common to all cases: a central issue (the pathology); the situation (symptoms); the background (patient history); and the documentation (patient file). The obvious approach to engineering a medical case study is to model a real patient file supplied by a doctor or specialist. This procedure can face several obstacles:

- For a patient's medical files to be made available, approval must be given by the medical review board of the clinic or hospital (even if the person's name and identification are removed). This procedure is not always quick or easy.
- Well documented case studies supplied by doctors or specialists often reflect their interests in exceptional, complicated and challenging cases. Rarely does this reflect the 'status-quo' sequela of a disease.

An alternative approach is to work backwards from the preferred end-point of a target pathology, one that is a result of intervention that meets the instructional objectives. Create a flow chart of the common symptoms, standard assessment and evaluation events that would lead up to the desired diagnosis and preferred treatment strategy. This approach keeps the interactive case study within the boundaries of the target content.

Q & A as an Instructional Tool

Quizzes do not always have to be limited to the function of an evaluation tool whose instructional value is limited to feedback and confirmation. Quizzes can also be constructed as instructional and remedial vehicles.

In the X-Plore series, one objective was to measure the users' existing knowledge, while ensuring that they reached a defined, measurable level of competence. This required a TOTE (test, operate, test, exit) loop that would not allow users to exit until they had mastered specific content points.

Because of disk size restrictions, there was no room for standard interactive instructional presentations. This challenged the interface to use Q&A as the main instructional tool.

The X-Plore instructional Q&A procedure was as follows:

- Section overview: One screen of text contained an overview of the information being tested in that section. A section overview contained several content points.
- Question overview: A sentence of content information preceding each question.
- The question and corresponding multiple choice responses.
- X-Plore: One or more supplementary content points added to the rationale for the correct response.
- The matching remedial 'inverted' question(s) at the end of each section which also included a question overview, question and X-Plore function.

In order to ensure that the section and question overviews did not provide direct clues to the solution of any of the questions, each was constructed to contain associated information relevant to each question's content domain.

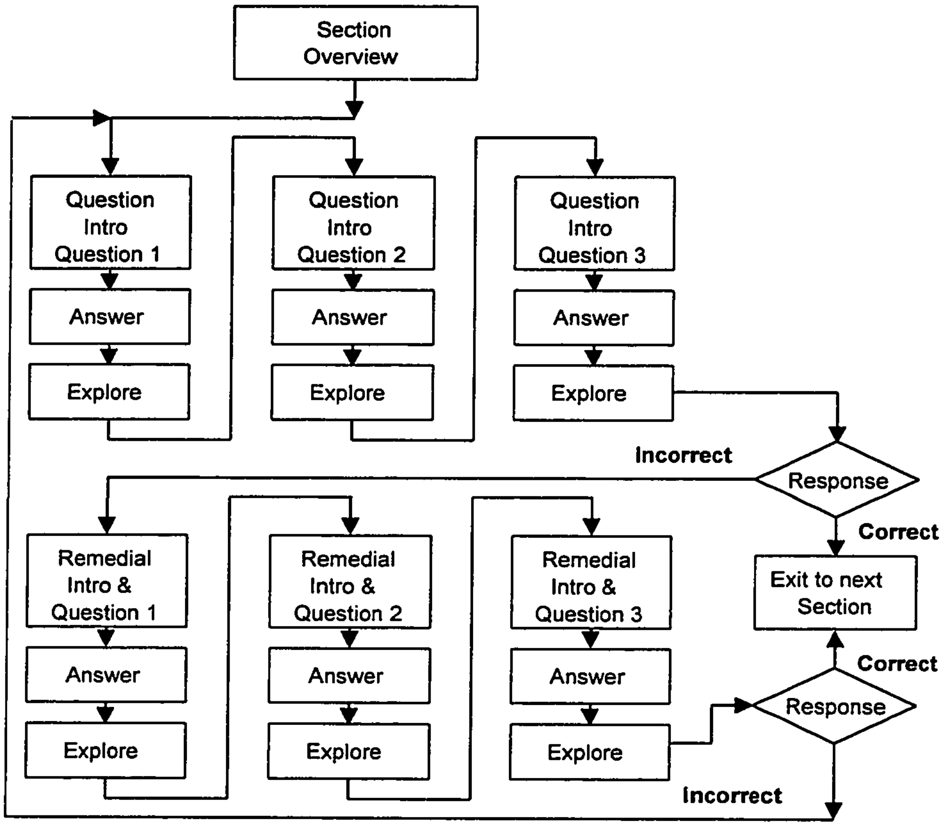


Figure 17: Q&A as an instructional tool with remedial loop in X-Plore series

Question Inversion

For the X-Plore series, the users' entry level had to be measured, and a 100% mastery of requisite content had to be ensured. To measure entry level knowledge, then guarantee that each user achieved 100%, the standard interactive design would proceed as follows:

- Direct the users through a pretest.
- Based on their performance, direct them to instructional activities.
- Direct the users through a posttest.
- Based on their performance, direct them to remedial instructional activities.
- Direct the users through a posttest, etc.

For the X-Plore series, this standard cycle presented two problems: there was insufficient disk space for standard remedial activities; and repetitive exposure to the same question was undesirable. If users repeated a question frequently, they would eventually get it right simply by short-term recall or process of elimination.

In the event that users did not get the first question right, the solution was to pair each question with a complimentary 'inverted' remedial question,. The following analogy illustrates the basic concept of 'question inversion':

- Original question: 3 x 4 = ?
- Inverted/remedial question: 12/4 = ?

In some circumstances, it is impossible to invert a question because of the content structure. In these cases, a question from the same content domain was used. To meet the requirement of mastery on certain content points, each incorrect answer was recorded. At the end of each section, a score was given and users were directed to areas within the supplementary print booklet for remedial work. For each incorrect response, users had to answer one remedial question based on the same content. If that question was incorrectly answered, users were referred back to the original question. This cycle continued until the question was correctly answered. The number of attempts to correctly answer each question was tracked and reported.

Step 5: Support Tools

Support tools are neither navigational nor key instructional tools, yet contribute to the overall effectiveness of an interface. Support tools can provide users with supplementary instruction and can assist them with tasks and exercises.

Support tools can be divided into four categories:

1. Administration (intro and exit screens, bookmark)
2. Resources, (hypertext, hyperlinks)
3. Feedback and reinforcement (confirmation, highlighting, audio, etc.)
4. Interface 'edutainment'

Note: With the exception of administrative tools, support tools tend to be optional in that users are not forced by the program to interact with them.

Administrative Tools

Intro and exit screens are classified as supplementary support tools because they have no direct instructional value. Positioned at the entry and exit screens of a program, they fulfill important reporting functions. These functions identify users and their frequency of use, collect data for reporting purposes, track users' progress for orientation and 'bookmark' functions, and perform several often overlooked legal functions.

Intro Screens

The function of intro screens (a.k.a. 'splash screens') is to provide:

- An engaging, attractive introduction that elevate the users' interests.
- Advanced organizers that advise users as to the subject, purpose and expected time in the program.
- A sign-in screen so that users can be:
  - identified in a multi-user environment;
  - identified in the client's database; and
  - returned to the location in the program where they left off (see 'Bookmarking' in the section on Navigation).
- An introductory path to basic navigation tools.
- A record of the users' frequency of use to determine their level of exposure to the navigation instruction loop.

In the X-Plore series, the intro screen only required the representatives' territory code (which at the reporting stage would access specific user data from the client database). In the Pacing Primer, the intro screen had to formally identify users for return mailing, and to differentiate them from other healthcare specialists in a multi-user environment.

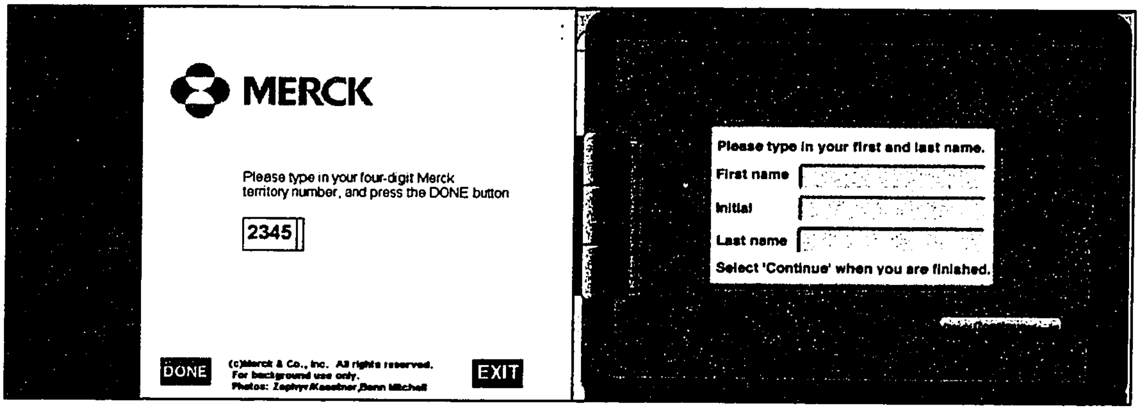


Figure 18: Comparison of intro screens from X-Plore series and Pacing Primer

Exit Screens

The function of exit screens is to provide:

- warning that users are exiting the program to return to the operating system, and confirmation that they wish to exit (with opportunity to return to the program if the selection is an error);
- reference to each user's last position within the program for the bookmark function (if a user has not completed the required content);
- a tracking/evaluation code, or notification of how results will be submitted if done automatically;
- special thanks;
- disclaimers and copyright clauses;
- e-mail or postal addresses, phone or fax numbers for comments or further inquiries; and
- credits for the producers, and any third-party install, authoring or viewing software, as required.

Bookmark

When users exit the interface to return to the host operating system, the program makes note of their last position within the program content/structure. When users log back on, the program bypasses all other 'intro' administrative functions, and returns them to exactly where they left off. This eliminates the need to manually advance through the interface to the last (known) position.

Resources

Resources provide adjunct information, or 'secondary' content (non-requisite) that enhances users' knowledge by rounding out comprehension, and providing peripheral or background details. Resources are also unique in that they are not part of the main or mandatory (forced) instructional stream/content. Users who enter the program with the prerequisite knowledge and follow the main instructional stream should be able to successfully complete a posttest, whether or not they took advantage of resource tools. Users who enter without the prerequisite knowledge will benefit most from the use of support tools.

On-line resources

The term 'on-line' refers to auxiliary databases, but there is some difference in opinion as to where that information resides. In some cases, if auxiliary information resides on the same CD as the host program, it is called an 'on-disk' resource. In other cases, 'on-line' refers to links that access information from a LAN/WAN, intranet or the Internet. In this discussion, resources retrieved from the host program disks/disc are referred to as 'resident' resources. Resources external to the install diskettes or CD-ROM are referred to as 'on-line'.

Access to on-line resources is often available via hyperlinks imbedded in the program. These can provide users with an almost unlimited resource base to enhance the learning experience. The medical interfaces in this discussion do not use on-line resources because of their following limitations:

- Connecting to an on-line resource requires that the host computer be part of a LAN/WAN or intranet, or have a high-speed modem for external access. This can reduce the minimum required install base.
- For the entire install base to have high-speed access, users would have to have accounts with a national Internet service provider with 1-800 access.
- Most interactive medical training products are designed to be used any time, anywhere. Access to on-line resources would require being connected at the time of use.
- Even with high-speed, national access, download time for anything other than text and simple images and graphics is still unacceptably slow (unless such national access was through ISDN, T1/2, cable or satellite links)
- Server load is greatest (and therefore access is slowest) during business hours and reasonable evening hours.
- Commercial medical databases such as Health Database Plus® and Medline® are pay services which require users to have an active account. Other databases are only accessible outside of business hours.
- Http and www sites often change URLs. The program then runs the risk of imbedding outdated hyperlinks that are no longer valid.
- Not all corporations have a resource database accessible within their intranet. Most intranets are used for communication purposes, or sharing, business-specific data.
- In the sensitive healthcare environment, all information must be approved by the training, medical and legal departments of the company -- an undertaking that would be near impossible for external resources. For this reason alone, most resources must be resident.
- Within the highly competitive pharmaceutical environment, many issues are sensitive. Therefore, access to proprietary data over non-secure systems unprotected by firewalls is generally discouraged.

Resident resources

Resident resources are instructional material and/or presentations that are not part of the main stream instruction. They are important components of 'learning on-demand', and improve the program's ability to accommodate a wide range of entry levels. The analogy (and icon) most often used is an encyclopedia or reference book which users select at their discretion. The amount of information that can be stored in a resource area is limited only by budget, production time, disc space and material relevance. The use of resident resources enables any program to offer a learning-on-demand component without Level Three interactive design.

In the Pacing Primer, the metaphor of a book lying on a table opened a resource area on electrocardiography that was as extensive as the core program itself.

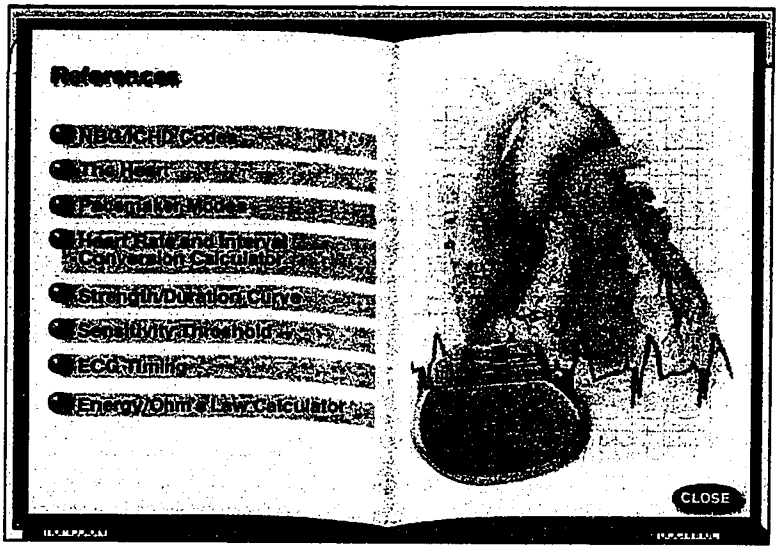


Figure 19: Index to Resource tools in Pacing Primer

Resident resources pose several problems that are unique to a learning-on-demand interface:

- Because resources are optional, a great deal of design and production work can go into a tool that may not be adequately explored by users, or possibly never used.
- Only motivated learners will exploit resident resources.
- Requisite content acquisition and evaluation does not force users to access resident resources.
- Mechanisms for directing users to the resident resources (usually for remedial purposes) have to be ever present and obvious, yet at the same time, unobtrusive.

Adjunct print resources

The issue of adjunct print material being included with interactive material has both opponents and proponents. Adjunct print material is useful in that:

- It reduces the amount of material in the interactive program itself. Not all content lends itself to interactivity.
- It reduces the size of the interface structure (and associated menus, and navigation tools).
- A book/booklet provides an extremely rapid random access database.
- All users know how to navigate print material. For some users, it is the preferred reference method.

The X-Plore series was shipped with manuals. This provided users with the basic content, since the interactive program itself was designed primarily to measure the users' knowledge base. Remedial questions also referred users to specific chapters and sections within these manuals.

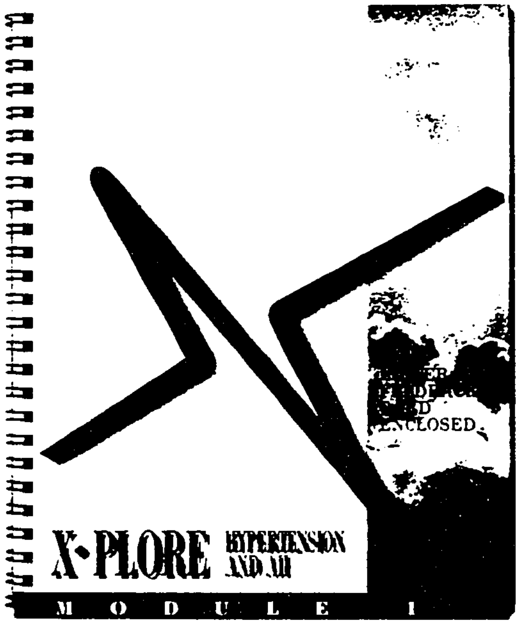


Figure 20: Manual from X-Plore adjunct print material

Sometimes adjunct text-only material can be made accessible from within the interface. Some advantages and disadvantages included:

- When the text is accessed, it reflects the interactive instructional material. (In the Backgrounder series, the text function constantly tracked the users' location in the content).
- Text can be read on a screen only 75% as quickly as on the printed page (Carliner 1990).
- Depending on the size and nature of the content, random access of text in an electronic document is not always faster than random access in a print document.
- Since a book icon is a de facto navigational tool standard for a dictionary, an alternative, non conflicting button has to be chosen.

The Backgrounder series included resident resource text in order to meet the needs of more traditional learners who were comfortable with text-based content. Attempts to design a 'Text' icon that would not conflict with the lexicon icon were not successful. Ultimately, a simple button marked 'Text' was included for accessing the resident print material related to the subject on the screen.

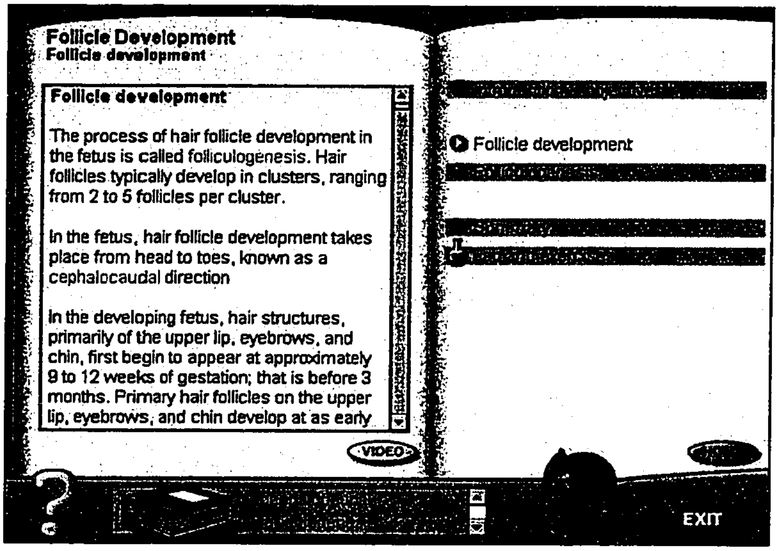


Figure 21: Text option in Backgrounder series

Hypertext

In its simplest form, hypertext provides users with the definition of the hypertext word. However, the definition of hypertext differs among authors. Mynatt, Leventhal, Instone, Farhat and Rohlman (1992) recognize that there is a shortage of conclusive results on the relative merits of hypertext when compared with traditional media.

Variations that increase the depth of hypertext include:

- audio pronunciation (by default or selected with an optional icon);
- supplementary text information beyond the definition (anecdotal, historical);
- supplementary tables, graphics, animation and/or video clips;
- HTML links to other areas of the program; and
- URL's to related web sites.

Hypertext is accepted as a mechanism for supplementary content, providing there is a limit to the extent of information available.

The hypertext balance

Information depth is a dual-edged sword. While it supplements and enhances the learning process, it also distances the learner from the principal instructional objectives. If hypertext windows offer additional choices, (e.g. hyperlinks, URL's, audio, video) users must proceed into additional levels, interrupting the learning stream while temporarily distancing themselves from the main message. Excessive hypertext also overburdens interface screens, increasing the signal-to-noise ratio, working against the effectiveness of the program.

A counter-argument to this problem is that users decide to access the hypertext depending on their requirements or understanding of the hypertext word or the associated concept (learning on-demand). However:

- Hypertext is not always the most appropriate and effective medium for supplemental information.
- If users are constantly overwhelmed with secondary information each time they select a hypertext word, they may avoid using the hypertext function.
- Excessive content dilutes the usefulness of the hypertext function.
- Ensuring that all hypertext is consistent in terms of the type, depth and quality could be equally overwhelming for the writers and designers.

The depth of hypertext must be carefully balanced within the interactive program because of visual, aesthetic, navigational and instructional impact. It is easy to get carried away with the potential of the hypertext tool, and as a result, carry away the users as well. Hypertext should be employed as supplementary support only. It should never be germane to the content delivery, nor the principal material on which users will be tested at the end of the program.

Hypertext mechanisms

Click-on/Click-off Pop-ups

This mechanism requires that users click on a hypertext word (usually identified in the colour blue) which opens a pop-up window with the definition. Users must then click on a 'Close' or 'Done' button, or on the interface background, to exit the window. This method requires an additional two 'inputs' (mouse clicks or keystrokes) to enter and exit the definition box. One alternative is an automatic 'time-out' that closes the box after a preset delay.

Rollover Pop-ups

Rollovers automatically pop-up the definition box whenever the cursor is moved over the hypertext. While this solves the additional input problem of click-on/off pop-ups, it creates another problem. Whenever the cursor unintentionally rolls over a hypertext word, distracting windows pop-up and disappear. As a result, the interface design should avoid positioning text with hypertext rollovers in the path that lies between instructional and navigational tools.

In simple screen layouts, pop-up definition boxes can be positioned anywhere, providing they do not cover the sentence in which the hypertext word resides. Depending on the size and content of the definition box and the layout density, larger definition boxes can cover other material on the screen,. Also, different screen positions, sizes and shapes of definition boxes (to avoid covering screen material) add to layout and programming workload.

Dedicated Definition Box

Depending on the volume, frequency and importance of hypertext as a support tool, screen layout should assign a dedicated hypertext area, wherever possible. The Pacing Primer defined a sector at the top of the interface screen for exclusive use by the hypertext. While this reduced the useable screen area by 10 to 15% percent, it resulted in a less obtrusive hypertext presence.

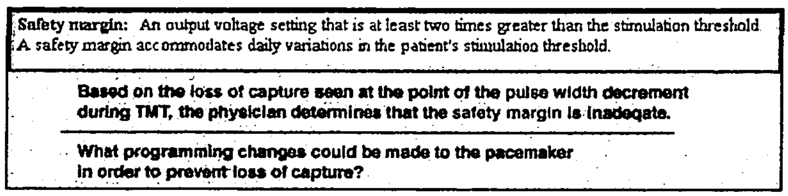


Figure 22: Dedicated hypertext area in Pacing primer case study

Coloured hypertext words within a text, and their definition boxes, can also be distracting. The Backgrounder series addressed this problem by removing hypertext and integrating a scroll-box into the bottom of the interface screen.

Scroll Box

A scrollable lexicon at the bottom of the interface screen solves the problem of definition boxes that cover on-screen material. Users search the lexicon for a word then click to open its pop-up definition box. The liability of this mechanism is that it reduces the useable screen area by 10 to 15% percent.

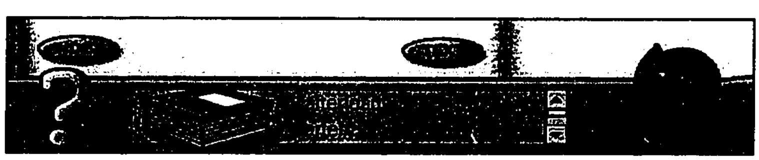


Figure 23: Scroll box in Backgrounder series

One final problem with pop-up, rollover and scrollable hypertext is that definitions are only available for words on the current screen. What if users would like to explore or confirm a definition of a word not currently displayed on the screen?

Searchable

One solution is a searchable hypertext protocol which responds to Lansdale and Ormerod's (1994) suggestion of linked content that enables users to access and view text-based material at random. This principle was applied in the Hypertension Backgrounder interface which, in addition to the definition scroll box, enabled users to open a definition search screen. While displaying the definition of all words in the current text, this screen also enabled users to search the entire lexicon with a simple search engine that did not require keyboard entry and spelling accuracy.

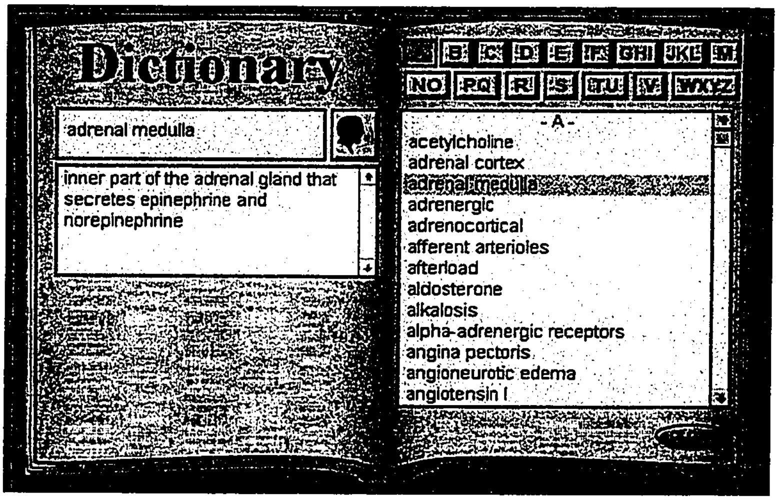


Figure 24: Searchable lexicon in Backgrounder series

Tips: Defining hypertext lexicons

Once the hypertext mechanism, access and positioning standards have been established, several issues remain regarding the content of the hypertext lexicon itself, and which words within the instructional text should be hypertext words.

- Which areas of the interface will contain hypertext help?
- Which words/how many words will be hypertext?
- What is the protocol for positioning repeat hypertext words?
- What will be the maximum number of characters per hypertext definition?
- When/should hypertext definitions contain supplementary non-text information such as graphics, photos, audio and/or video information?

Those new to the interactive platform often select excessive lexicons for inclusion. Text should be kept to a minimum on any screen, and excessive hypertext can become overbearing in sentences, therefore a hypertext application standard should be established and practiced.

The following guidelines are useful in defining a lexicon and in determining where hypertext should be positioned.

1. **Where:** Evaluate the use of hypertext in certain areas such as:

    - any test/quiz/question areas (definitions may reveal the answer);
    - any answers provided in test/quiz/question areas (remedial work should be done by reviewing the presentation material related to that answer);
    - any explanations for answers in test/quiz/question areas (Explanations usually evoke a pop-up box. Hypertext would add a pop-up to a pop-up);
    - other pop-up boxes (same rationale);
    - introductions (key concepts defined later in presentation areas); and
    - summaries (key concepts that should have been acquired earlier).

2. **Which words?**: Hypertext should be limited to words that are unique to the subject of the program. The best method to determine this is to:

   - Search existing material for a glossary unique to the subject. (E.g. for the Pacing Primer a Pacing Glossary was the starting point. Glossaries for the Backgrounder series were extracted from the original Backgrounders).
   - Combine all screen text that qualifies for hypertext into one e-file. Run a search through the file with all the words from the glossary, eliminating those words that do not appear in any qualifying text.
   - Submit the remaining words to the client and request that -- after verifying, correcting, rewording, abbreviating and approving the definitions -- they rank them ABC as follows:

     A) Absolutely, must be included.

     B) But it would be useful if these were included as well.

     C) Could be removed.
   - Remove all of B and C
   - Do a word count on your screen text file (screens that qualify for hypertext). If the definition-to-total-screen-word-count ratio exceeds 10%, start chiseling away at the A-list (Absolute).
   - Provide a sampling of users with a list of all proposed hypertext words and ask them to select the words they feel should be defined.
   - Test a second sampling of users with the definitions (using a fill-in-the-blank format or one similar to the *Readers Digest* 'Word Power' multiple choice format). Words that score poorly should be hypertext.

3. **Repeat words:** Some suggestions for a protocol for words that occur more than once in the program are:

   - If the word appears more than once in a sentence, hypertext only the first appearance.
   - If the word appears in the next screen (and that screen can easily be recalled), hypertext the word only in the first screen.
   - If the word appears in a screen two actions later (third screen), hypertext the word again.
   - Following the above, if the word continues to appear in following screens, it should not be hypertext again in that branch of the program.
   - If users can access program sections (branches) in a random order, and the word in question appears in another section (branch) of the program, the above guidelines should be applied.

4. **Definition size:** When definitions are being displayed in pop-up boxes with a defined area, it is important to limit their length. To determine their size:
   - Build the maximum size definition box on a test sample of the interface screen, sized for the delivery platform.
   - Fill it with characters of the chosen font and size, and count them.
   - Establish 85% as the character limit for fixed-position definition boxes.
   - This method is only appropriate for definitions that do not contain additional non-text material such as graphics, photo's, etc.

Hypertext Options

There are numerous options that can be added to enhance hypertext definitions, such as audio pronunciation, graphics, photos, and even video.

Audio Pronunciation

Audio definition replaces the standard phonetic text pronunciation. It removes pronunciation ambiguity, is relatively easy to program, and does not take up much space on a CD-ROM. In situations where the word is frequently used in the narrated presentation of a program, an audio definition may be redundant. In the Backgrounder series, the salesforce had to use correct pronunciation in their daily discussions with doctors. For those not familiar with the correct pronunciation of certain words, audio definition was an appreciated option.

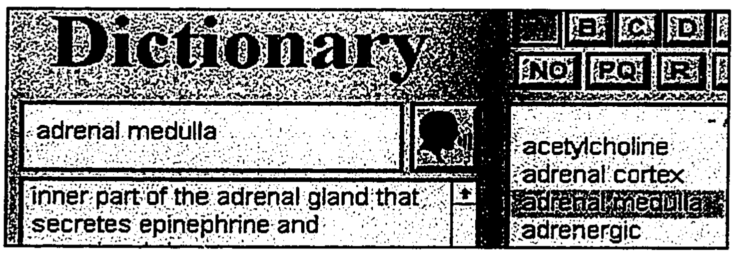


Figure 25: Audio pronunciation option in Backgrounder lexicon

Graphics/Photos

Appropriately placed, one picture may be worth a thousand words. The problem is space. One objective in positioning hypertext is to keep the screen space it occupies to a minimum. When planning to incorporate graphics or photographs:

- Take all potential images into consideration when planning the size of a fixed position or elastic definition box.
- Graphics should be simple line drawings with no gradation or labeling text. Photographs should be close-ups with limited colour and detail.
- *Keep definitions to a minimum to allow for positioning of the image.*

In some cases, a graphic can be included where it helps in the understanding of the definition, and is not available in any other part of the program. In the Backgrounder series, if a graphic was important enough, it was included in the presentation portion of the program. In the Pacing Primer, any graphic that merited inclusion was placed in the resources.

Video

It is also possible to imbed QuickTime, AVI and/or Director movies into hypertext. None of the interfaces used as examples here have included hypertext video. However it does raise the issue as to what depth is necessary to provide supplementary information on-demand.

Note: Mynatt, et al (1992) conclude that the advantage of hypertext dictionaries over other electronic text formats cannot be attributed to non-linearity, linking or other hypertext-specific properties. However, for questions which required the location of detailed text content, it proved to be superior.

Hyperlinks

"Links are a distraction. It is pointless to write a paragraph and then fill it full of invitations to your reader to go elsewhere."

Lynch and Horton (1997)

The next logical step beyond hypertext is hyperlinking. Hyperlinks enable users to 'jump' to another related area of the program. Designers considering this option should keep in mind the risks of enabling users to 'step aside' from the main stream presentation, possibly becoming disoriented or lost within the interface. Lynch and Horton (1997) observe that links are a distraction that detract from the main content while adding to the length of delivery.

An interface employing hyperlinks requires a universal 'Back/Forward' function. Fortunately, most navigation tools for advancing or reviewing screens, presentations, case studies, etc. have a 'Back/Forward' function. However, two similar tools providing two different functions could create some navigation confusion.

In the design of the interfaces discussed here, hyperlinks were excluded in favor of standard Back/Forward movements within a limited number of levels.

Interface Edutainment

Edutainment makes instructional material engaging and enjoyable and gives a program a certain amount of character. In a study of 11 major organizations, Oz and White (1993) reported that a level of 'built-in entertainment' helped maintain trainees' attention. These relevant, but peripheral devices demonstrate that it can be advantageous to step away from strict interface design convention in order to provide users with something unsuspected or out of the ordinary.

Random Events

Random events that are integrated into the program structure provide an element of surprise and unpredictability, humanizing the program.

In an earlier X-Plore series, two random events were introduced into feedback mechanisms. A character, 'Molly Cool' (Molecule), would appear randomly in screens as positive feedback (with the expression 'Cool'), reinforced by a simple gesture such as a wink, smile, thumbs-up or 'high-five'. Responses to correct or incorrect answers to questions were phrased as 'Great', 'Excellent', 'Genius' or 'Oops', Too bad', or 'Almost' -- drawn at random from a response database.

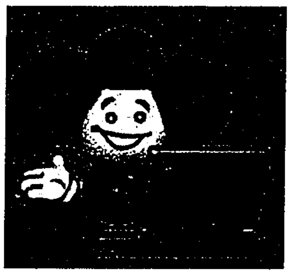


Figure 26: Random event in X-Plore series

Hidden Screens or Menus

Hidden screens or menus are usually imbedded and launched from main menu areas. The trigger mechanism is an object in the interface that is not an obvious navigational or instructional icon, and is not identified. The value of these devices is that they reward curious users with 'surprise' and useful information. This encourages users to explore all areas of the interface and not to limit themselves only to the main menus.

In the Pacing Primer, an innocuous 'bulletin board' in the main interface rewarded curious users with supplementary information that -- although not part of the principal program content -- was of general interest. Hidden screens or menus are also good 'parking lots' for valuable, adjunct information that does not fit into the program structure or instructional presentations.

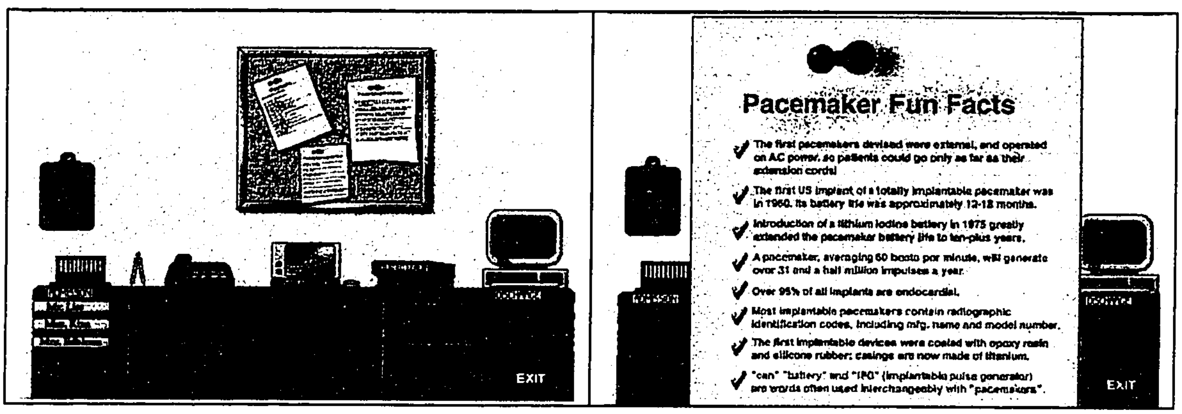


Figure 27: Hidden menus (bulletin board) in Pacing Primer main menu

Step 6: Structure

"The goal of most organizational schemes is to keep the number of local variables the reader must keep in short-term memory to a minimum, using a combination of graphic design and layout conventions along with editorial division of information into discrete units."

Lynch and Horton, 1997

Once the metaphor and learning strategies have been integrated and their respective instructional and support tools selected, it is necessary to provide users with a structure that positions all tools physically, sequentially, by hierarchy and relation. This provides users with a logical means of exposure and access.

Structure should be designed taking into consideration the users, the content, the metaphor, and instructional and support tools.

The Users

If the users are relatively familiar with navigating commercial software or interactive courseware, there is less chance that they will get lost in a 'thick' interface structure. Conversely, if users are relatively inexperienced with such tools, a simple, 'thin' structure is preferable. The X-Plore series was designed to introduce users to interactive training, therefore the structure was very simple. As users progressed into the Backgrounder series, the structure became more complex.

The Content

In many cases, the hierarchy and structure of the content dictates the structure of the interface. If the content has many levels in its natural hierarchy, and the program objectives require that the designer cover all the material contained within the hierarchy, the program structure will reflect that hierarchy. For example, the modules, sections and menus of the Backgrounder series reflected the organizational hierarchy of the original print material.

The Metaphor

The metaphor should be able to support the structure. The Backgrounder 'classroom' supported the five modules (books on shelf), each module supported up to four sections (chapters in a book), and each section supported up to five presentations (pages in a chapter).

Instructional and Support Tools

The structure must accommodate the instructional and support tools which are the common interface elements that bind the content together, enabling users to move effortlessly between them. The Pacing Primer nursing station structure enabled users to access content (resource book) and support tools (clinical evaluation equipment) while exploring the case studies (patient files).

Lynch and Horton (1997) offer a general four-step approach to structure:

1. Divide content into logical units.
2. Establish a hierarchy of importance and generality.
3. Use the hierarchy to structure relationships (navigation) between chunks (instructional and support tools).
4. Analyze the functional and aesthetic success of your system (does it 'fit' and make sense?).

Maul and Spotts (1993) identify common errors that detract from a structure's effectiveness.

- Incorrect (content) branching.
- Inconsistent content organization.
- Lessons (presentations) that are too large/long.
- Too many or too few menus (modules, sections or topics).

Tip: A quick structural snap-shot

When considering metaphors, instructional strategies and support tools, it is useful to have a general idea of what the structure of an interface may look like, before detailing the structural tree.

1. Using a flow-chart program, break the content down into its natural information hierarchy. This provides a visual map of the program structure.
2. Identify and segment modules, sections, topics, and individual screens around the branching/layers of this flow-chart.
3. Add intro/overview, and menu screens at all branching points.
4. Add summary screens at the collective conclusion points.

Given these considerations, both the structure and the method of construction of each interface will always vary along with the style of the designer. This too adds to the 'character' of an interface.

Levels and Menus: depth vs. breadth

The issue of levels and menus is a dual-edged sword. On the one hand, a multi-layered interface often enables a more exact structuring that mimics the natural hierarchy of the content, if there is one. However, the more levels there are, the 'thicker' the interface. The liability of a 'thick' interface is that it distances users from the content and increases 'navigation' time in proportion to 'learning' time. Users spend more time getting to where they want to go, and less time exploiting the benefits of the interactive learning environment. The interface designer often has to determine what is preferable: depth (less choices per menu, but more levels), or breadth (more choices per menu, but less levels).

Levels vs. Menus

Level and menu limits are generally accepted as:

- 3-4 levels (Shneiderman, 1987)
- 4-8 menu items per level (Shneiderman, 1987)
- 5-7 menu items per level (Lynch & Horton, 1997)[7]

The obvious advantage of depth vs. breadth it is that depth can cover more information than breadth, as illustrated by the following table: (modified Lynch & Horton, 1997)

---

[7] These figures are congruent with Miller's (1956) studies of information chunking limits of 7, +/- 2

choices / menu	2 levels	3 levels	4 levels	5 levels	6 levels
2	4	8	16	32	64
3	9	27	81	243	729
4	16	64	256	1024	4096
5	25	125	625	3125	
6	36	216	1296	7776	
7	49	343	2401		
8	64	512	4096		

Table 11: Relationship between structure and total number of screens

For example, three choices per menu on five levels (3:5=243) offer almost twice the number of screens as the converse structure of five choices per menu on three levels (5:3=125).

However, several studies have shown that breadth is faster, more accurate, preferable, and easier to learn:

- Because of a 'significant order effect' a one-level menu is easier to learn than a two-level menu (Dray, Ogden & Vestewig, 1981).
- A deep narrow structure of two items per level on six levels (2:6=64) produces the slowest, least accurate and least preferred version.
- A short broad tree with eight items per level on only two levels (8:2=64) is the best for speed, accuracy and preference (Miller, 1981).
- Search times are almost twice as long for a tree with two choices per level on 12 levels (2:12=4096) as opposed to 16 choices per level on a three level tree (16:3=4096) (Doughty & Kelso, 1984).

MacGregor and Lee (1987) offer a formula to estimate individual search times, to assist in comparing structures and selecting the one which is more efficient.

Search Time = p{E(A)t +k+c}

where: p = number of levels
E(A)= number of options per menu
t = time to read a single option
k = time to make a selection
c = computer response time to display new screen (for video)

Based on an optimal model of this formula, MacGregor and Lee (1987) recommend 4-8 items per menu to minimize search time. The trade-off is that a higher choice per level ratio means less options.

What all this suggests is -- as with many issues in interface design -- breadth vs. depth is a yin and yang issue where the designer attempts to strike a healthy balance that best suits the users, content, and instructional objectives.

	Breadth	⇔	**Depth**
Characteristics	few menus, many choices		many menus, few choices
Advantages	faster, easier		more information
Disadvantages	screen crowding decreased separation		distance from core content disoriented or lost

Table 12: Comparative advantages and disadvantages of menu structures

Although structure often accommodates the natural hierarchy of the content, it is preferable to keep the interface structure shallow. Use as many items per screen as the interface area will comfortably accommodate.

In the X-Plore and Backgrounder series interfaces, maximum structure depth (as measured from the main menu) never exceeded three levels (including hypertext definitions). The same applied for the Interactive Pacing Primer (with the exception of the Resources, which brought the number of levels up to four or five depending from which level the resources were accessed). Menu items in the Backgrounder and Pacing Primer series were limited to five, (with the exception of the Resources area which offered up to eight items).

Sequencing vs. Branching

One technique that keeps an interface 'thin' is to avoid numerous decisions that either branch-out extensively, or layer the interface. Content should be sequenced into short digestible presentations (chunks), or structured via an index (which adds only one layer or branch), offering a selection of short digestible presentations. Either way, these presentations should include frequent stops for an exercise, lab, or just a break. Carliner (1990) recommends meaningful interaction every three to five screens.

The X-Plore series employed a simple linear structure with a hierarchy of 2x4x5 (modules x sections x transactions).

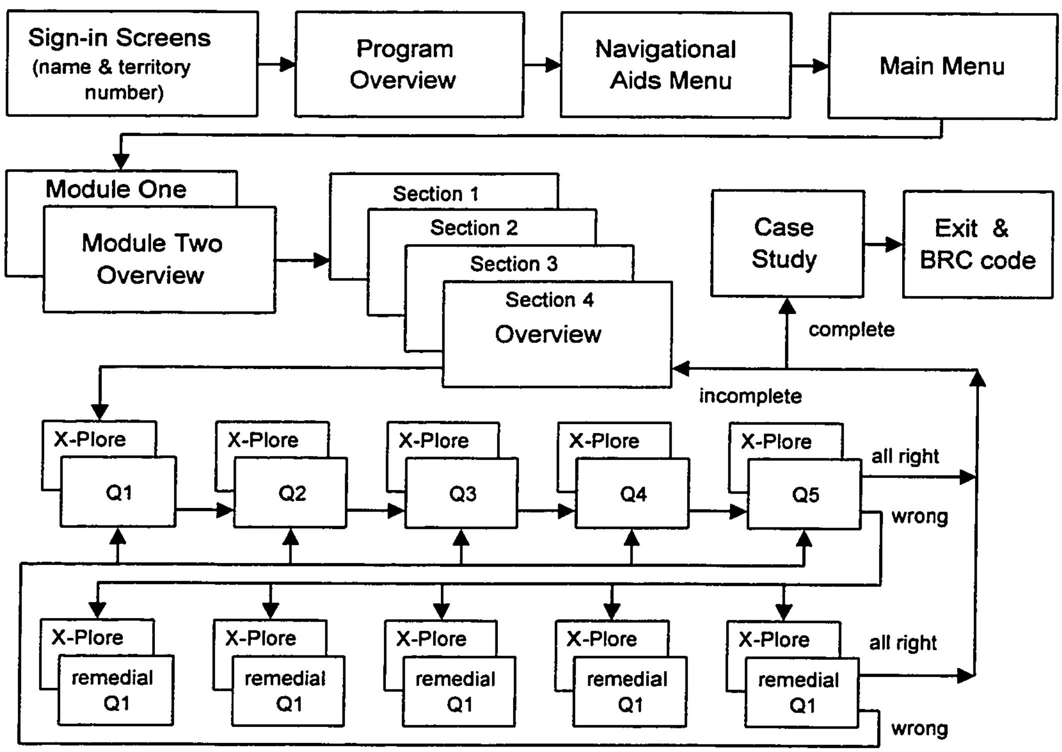


Figure 28: X-Plore structure

The Backgrounder series was also built on a fairly standard structure. On average, the presentation component was constructed with a core hierarchy of 5x4x5 (modules x sections x presentations) excluding objectives/overviews and summaries. The Lab component was constructed with a parallel hierarchy of 5x4x1 (modules x sections x dynamic models or case studies) excluding the lab exercise component.

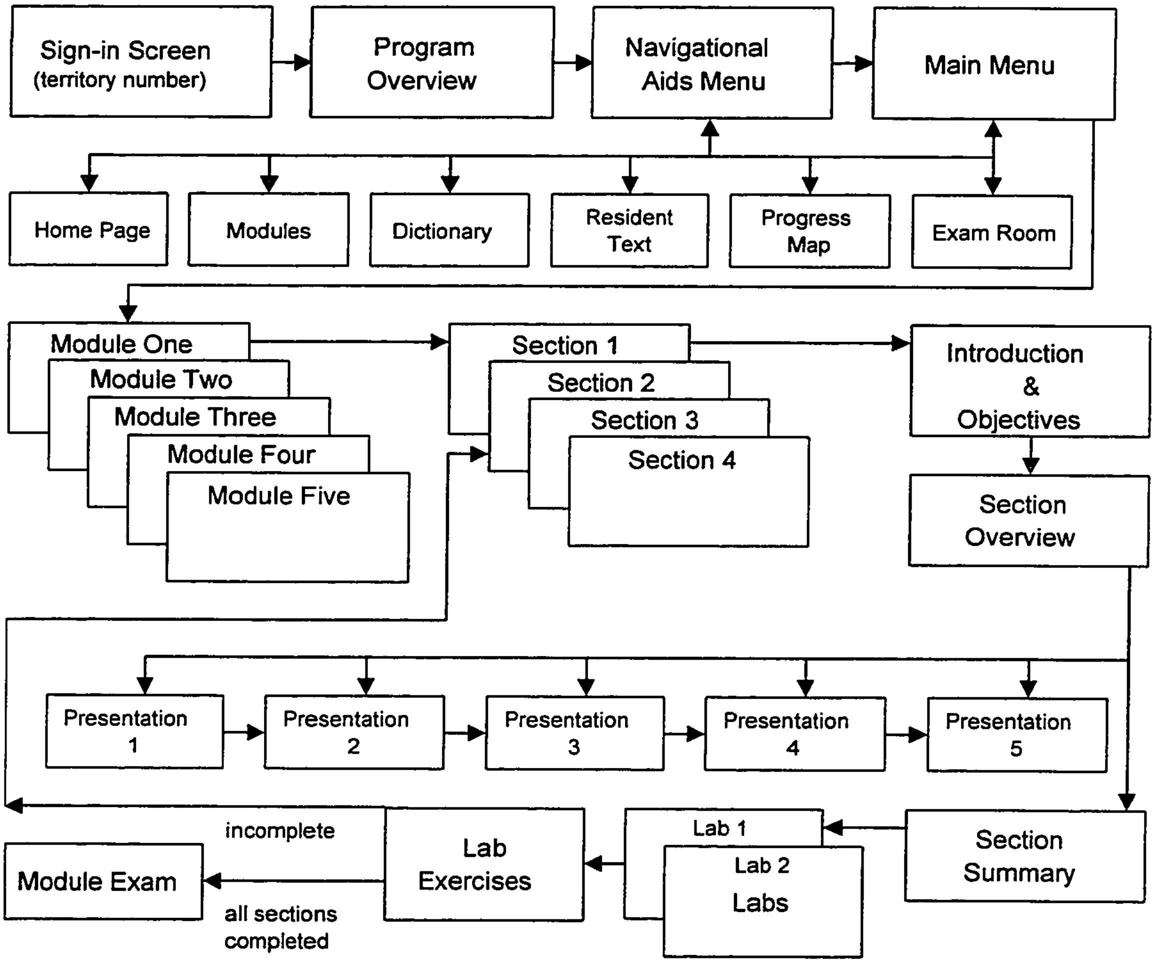


Figure 29: Backgrounder structure

The Pacing Primer series employed a multi-leveled structure in which various reference and clinical resources were available from within the core case study structure. On average, the case study structure was 3x10 (patients x assessment steps), the clinical resource structure was 4 x 1-6 (resources x one to six options), and the reference resources structure was 8 x 1-3 x 1-11 (subjects x one to three options x one to eleven dynamic models, calculators or demonstrations).

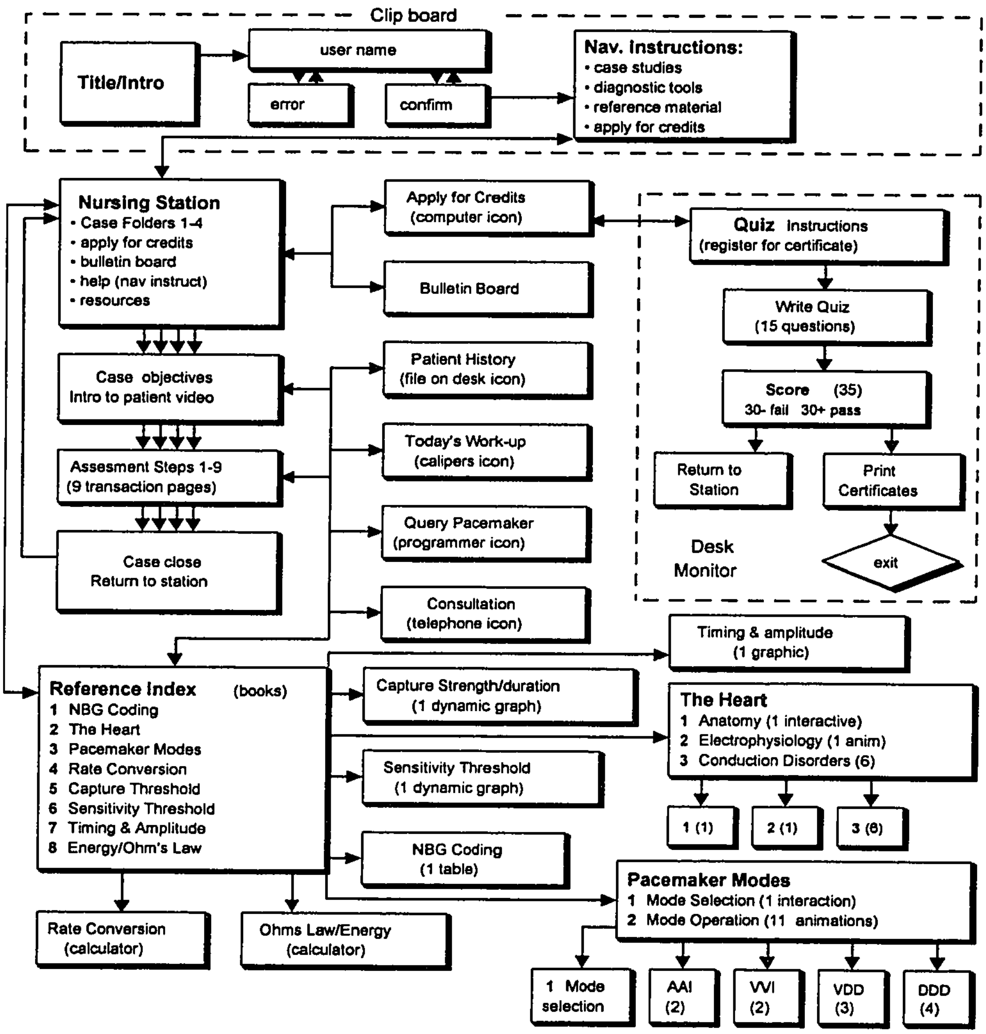


Figure 30: Pacing Primer structure

Assessing Structure

The structure is the backbone of an interface and bears some responsibility for the spirit of the metaphor while hosting the instructional tools that deliver the content. There are numerous ways of building structures to achieve the same instructional objectives. How does the designer determine which one is best for each application? One method is to run any proposed structure through a few fundamental 'acid tests'.

The four tools that can assist in designing navigational structures, or choosing between several possible designs are: time-on estimates, 'navigation-to-interaction ratios', screen-to-layer ratios, and reciprocity ratio.

Time-on Estimate

Simple but useful, time-on estimates calculate the total time that users will spend within the program, given a particular structure. This estimate is also useful in meeting the target program duration.

1. Using the structural flow-chart, assign an estimated interaction time to each screen by timing (or estimating) the:
   - read-time of any on-screen text;
   - run-time of any audio (or by reading the narration script);
   - run-time of any video sequences;
   - viewing time for graphics (e.g. time required to read each label); and
   - interaction time for each exercise or dynamic model.
2. Add up the total time. Assuming all other elements are equal, whichever structure is substantially shorter in global time-on is the preferred structure.

Navigation-to-Interaction Ratios

Navigation-to-interaction ratios help determine if there is an excessive amount of navigation given the content volume.

1. Using the structural flow chart, count the max./min number of inputs (e.g. mouse clicks and/or keystrokes) required to navigate from one area of content to another, and to interact with the content.
2. The preferable structure is one which has the least amount of user input for navigation, and the greatest amount of user input for interaction. It is important to avoid a syndrome that Lynch and Horton (1997) sarcastically refer to as "300K of drop-dead graphics, .01K of navigation options." (Section 5 pg. 34).

Screen-to-Layer Ratio

Using the structural flow charts, tabulate the amount of screens in the entire program and the number of distinct layers, branches or sections.

A high ratio of layers-to-screens generally suggests that users will spend more time interacting than navigating, although it does not necessarily mean the structure is superior. A low ratio of layers-to-screens suggests that users will spend a greater proportion of time navigating, compared to a structure with a higher ratio of layers-to-screens.

Reciprocity Ratio

The reciprocity ratio measures the number of actions initiated by users, compared to the number of actions initiated by the program. A high reciprocity ratio suggests a user-centered locus of control. A lower ratio suggests a lower level of interaction.

Step 7: Navigational Tools

> *"...there is a role for training of skills for interface search... training people how to learn, and is likely to become an increasing requirement as the burden of instructional design is placed more on interface designers rather than on trainers."*
>
> Lansdale and Ormerod (1994)

Once a structure has been constructed, it is necessary to select and design devices that will facilitate the users' exploration and use of various instructional tools. However, there is an inherent disadvantage when designing navigational tools for interfaces destined for limited-use and narrow audiences with limited experience. With the type of interfaces discussed here, users do not interact with the interface long enough to merit a highly detailed, and/or layered interface. The large investment of time required to learn commercial software interfaces is justified when users could spend several years using that tool.[8] The situation is the exact opposite for limited-use, narrow audience tools.

The Pacing Primer interface was designed to be used only once or twice. The design took into consideration that further training packages might not be developed for that segment of users, or that additional training packages might not be developed for that subject area. In the case of the X-Plore series, the average time-on was designed to be very short. As a result, the interface was very light with very few navigational tools to learn. In the case of the Backgrounder series, the interface was used not only over several CD-ROM's in each series, but over several series addressing three different pathologies or conditions. Therefore the Backgrounder interface could afford to be more complex, offering more learning strategies supported by more navigational tools.

It should never be assumed that the time spent learning a more 'complicated' interface can be amortized over many programs because:

- The first program in the series should not require that users spend a disproportionate amount of time learning an interface.
- New hires may not be exposed to previous programs in the series.
- Series can end prematurely.

With this in mind, the designer must consider:

- What type of input device should be favored.
- When and how menus should be integrated into the navigation scheme.
- When and how direct manipulation should be integrated into the navigation.
- How common navigational tools can best be used.

Once fundamental navigation tools have been chosen, the following questions can be answered.

- Do the tools lend themselves esthetically to the interface analogy?
- Are the tools intuitive? Will users recognize their implied function?
- Are there as few tools as possible?
- Are there as few levels behind each tool as possible?
- Is each tool multifunctional so that when users learn the use of a tool, that investment is amortized over numerous functions across numerous screens. (E.g. 'Done', 'Exit', 'Back' and 'Forward' arrows, and 'Help' buttons).
- Is there sufficient separation between the navigational tools to avoid cross-talk. E.g. is the functional difference between 'Exit' and 'Done' evident?
- Are the tools 'foolproof'? Can users do things that might have unintentional results, require effort to reverse, or that may discourage them?

---

[8] e.g. MS Word 6.01 has eight key pull down menus with an average of 15 sub-menus. At only the second layer (which addresses only a fraction of the programs capability), the user must understand the function of 120 operators.

An understanding of the users and their experience with other interfaces is the guiding principle in selecting navigational tools. This information can be used to determine the complexity of the tools, the duration/frequency of their use, and the depth of the corresponding menus. Otherwise, as the interface grows in complexity during the design phase, so will the navigational tools.

User Input Devices

> *"In the grammar of interactive media, buttons are the most basic cognitive element."*
>
> Goldberg (1996)

There are many user input devices available to the interface designer such as: keyboards, pointing devices, kola pads, touch screens, LCD stylus screens, and voice recognition. However, the largest general input device install base for medical training products at present is the standard WIMPS (Windows, Icons, and Menus Pointing Systems) equipped with the standard keyboard and pointer (usually a mouse), or in the case of a notebook computer, a pointer, track-ball or touch-pad.

In this standard configuration, users navigate the interface through three input options that determine the primary interactive style: menu selection, keyboard command, and direct manipulation.

When selecting one or more input devices, consider that the ease of learning a program depends on the users' ability to associate an input action with the program's reaction. This is achieved with two forms of memory:

- Recognition memory: Users associate a screen object with a function, effectively recognizing what it will do.
- Recall memory: Users recall skills from long term memory in order to manipulate the interface (e.g. writing a word, typing in a value, using a shortcut alpha or numeric command).

Most navigation and interaction for the X-Plore, Backgrounder and Pacing Primer interfaces was through menu selection or direct manipulation using a pointing device. Programs that use recognition memory are generally easier to learn than programs that require recall memory (Brown and Cunningham, 1989). This was an important consideration given the limited exposure users have with the interactive programs such as those modeled here because:

- Not all users have high-level keyboard skills.
- Type-in responses make syntax errors possible, which can frustrate users.
- Type-in responses require word recognition, which is more complex and time consuming to program.
- Since the majority of users are uni-dexterous and manipulate a pointing device with the preferred typing hand, constant shifting between keyboard and a pointing device slows down interaction time.

Tip: Selecting an input device

Because input device selection impacts the entire interface design and subsequent navigational tools, it must be considered as part of the interface design process. Discussions as to what input devices are most appropriate will vary according to subject, platform, objectives, audience, time constraints, environment, etc. At present, the most common input devices are the keyboard and mouse. The question that remains is "what input methodology is most effective for instructional, supplemental and navigational tools in this context?" There are three possibilities: keyboard only, pointing device only, or both.

Card, Moran and Newell's (1983) goals, operators, methods and selection (GOMS) model, and keystroke-level model (KLM) provide the following timing:

Action	Timing
Mental operation*	1.35 sec.
Keystroke	0.28 sec.
Pointing to target on small display with mouse	1.1 sec.
Moving hand from keyboard to mouse	0.4 sec.
Moving hand from mouse to keyboard	0.4 sec.

(*The time it takes to retrieve information and move from LT to ST memory)

Table 13: Input device timing

Using these timings, consider three input operations. In each scenario, a user reads a screen; reacts using an input device; the screen changes; and the cycle repeats itself for a total of three screens and three user inputs. Since it is a constant, the time required to read the screen, and the screen refresh delay, is not included. Using the above timings and cycle, the total execution times are:

Action	Timing (sec.)
Keyboard only	3.26
Pointing device only	4.9
Mouse and keyboard (typist)	4.88
Mouse and keyboard (non-typist)	6.51

Table 14: Input device variables timing

As expected, keyboard entry is the fastest option, assuming several conditions:

- The user is a skilled typist.
- All interface reactions could be achieved with only a keyboard, which will exclude direct manipulation as a navigation tool.
- The command requires a single keystroke (most software use two or more).
- The user is familiar with the interface, has learned the keystroke command, and doesn't require a second mental operation to locate and retrieve the correct command sequence.

'Mouse only' and the combination of mouse/keyboard input had similar operation timing. However, with the interactive interfaces discussed here:

a) direct manipulation was required;
b) the average user was not a trained typist (therefore 'hunt and peck' typing would double the timing); and
c) there could be no assumption that users would have had previous exposure to that interface... and would be familiar with keyboard input commands.

Given condition (a), keyboard only input was not an option. Given conditions (b and c), the mouse/keyboard input timing would exceed a mouse only-protocol. Therefore, a pointing device was favored for the X-Plore, Backgrounder and Pacing Primer interactive tools. One exception was where keyboard input was evidently more effective (logging in, type-in responses in labs, etc.). Also, certain keyboard inputs such as 'Return' were allowed for commands such as 'Done' and 'OK', since this is reflexive in certain response situations (e.g. dialog boxes).

Selecting the appropriate input device underscores the importance of researching the install-base, and knowing the users, content and instructional strategies when constructing interface tools and protocols that will best suit the users' needs.

Menus

"Perhaps the most common machine led dialogue is the menu"
Lansdale and Ormerod 1994

Menus are the de facto tools for navigation where there is more than one path branching from any one point in the interface structure. As fundamental as they may be, there are menu standards which should be observed. General rules-of-thumb for menus include:

- Arrange according to logical groupings and sequences.
- Use short, descriptive, interesting and/or revealing titles.
- Be consistent in name and design.
- Use novel selection mechanisms and devices judiciously.
- Conform to interface theme and screen area.
- Keep distinction between menu items and navigation tools.
- Limit the number of menu items because as that number increases:
  - distinction between menu items decreases;
  - it becomes more difficult to ensure correct category and hierarchy; and
  - screen area decreases.
- As the number of items in a menu decrease:
  - the number of menu's and levels increases; and
  - selection becomes less specific.

Menu Types

The interactive tools discussed in this thesis used fixed menus (all information present). However there are several other menu display options:

- pull down (or out L-R, or up) which is held by the cursor;
- pop-up (similar to dialogue box);
  - roll-over
  - click-on/click-off
- scrolling can be added to both pull down or pop-up; and
- parallel dialogue style can be added (use pointer or key command).

In the X-Plore, Backgrounder and Pacing Primer interfaces, fixed menus were the protocol of choice. Although pull down, pop-up, scrolling and parallel dialogue menus add more flexibility, they add to the complexity and introduce inconsistency. When a screen requires additional interactive devices for the sole purpose of accessing menus, the designer should consider re-evaluating the structure.

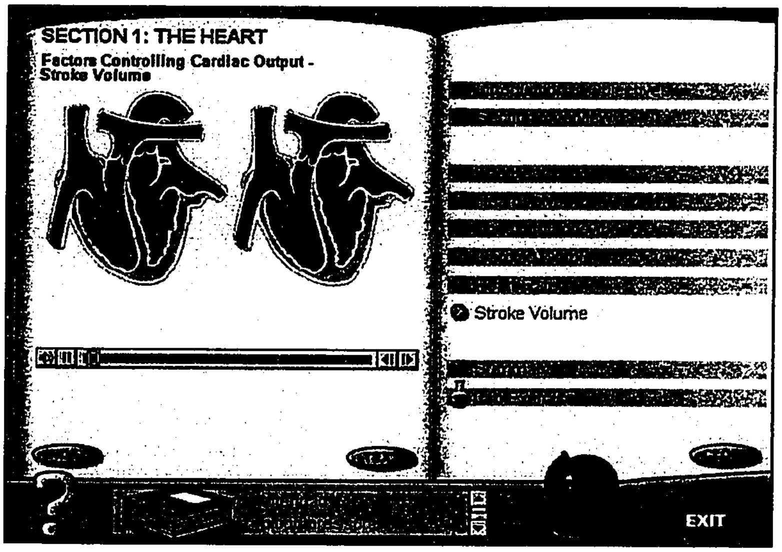


Figure 31: Menu from Backgrounder

Direct Manipulation

The physical manipulation of objects within the interactive screen has a variety of advantages beyond 'interactivity'. Shneiderman (1990) recommends direct manipulation because:

- Novices can learn functionality quickly.
- Experts can work rapidly over a wide range of tasks.
- Knowledgeable intermittent users retain operational concepts longer.
- All users see an immediate reaction and progress.
- Users have a sense of control.

For these reasons, the Backgrounder series and Pacing Primer interfaces exploited direct manipulation of objects within the interface, whenever appropriate.

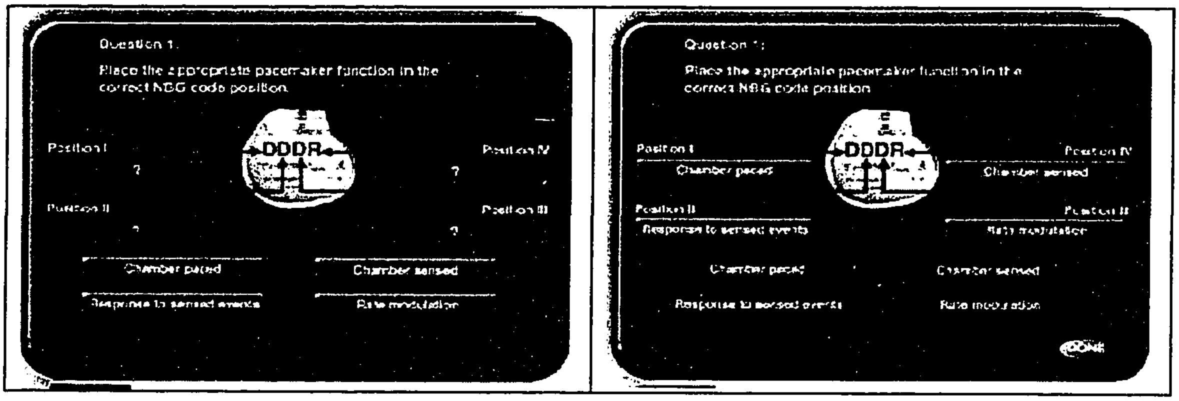


Figure 32: Drag and drop direct manipulation in Backgrounder

The following is a checklist of specific properties that should be designed into direct manipulation (modified Shneiderman, 1987):

- Explicit and visible action (visible and/or audible confirmation).
- Immediate feedback (users know the system is responding or assume error).
- Incremental effects (break complex tasks into simple, sequential steps).
- Intuitive interaction (emulate users' 'model' of task).
- Hierarchy of complexity (confirmation of mastery at each level).
- Pre-validation (only active choices should be present. Inactive objects or selections that are not valid should be 'greyed-out').

Common Tools

There are numerous issues surrounding the navigation tools that are common to any interface. 12 fundamental tools used in most interactive interfaces include:

1. Help tools and navigation instruction
2. Return, Enter, OK and Exit
3. Next, Back
4. QuickTime, AVI or Director movies
5. Scrolling
6. Pop-up vs. dialogue boxes
7. Standardized tools (from MS and Mac OS)
8. Icons
9. Orientation tools (mapping)
10. Confirmation/feedback (positive click, sound, etc.)
11. Transitions
12. Print navigation instructions

Help Tools and Navigation Instructions

Help tools are a redundant function incorporated into a program because the interface is not sufficiently self-explanatory. In principle, a user should not require any help with a well crafted interface. But in the real world, a balance must be struck, especially when designing interfaces for limited use where users will have short term exposure to the program.

Limited Help	Comprehensive Help
highly capable users	technophobic users
users exposed to earlier interfaces	new hire
recycled interface analogy and tools	new interface analogy and tools
simple, thin structure	complex, deep, wide structure
single learning strategy and tool	many learning strategies and tools

Table 15: Classification of two levels of help

Navigation instructions also reflect the simplicity or complexity of the program. For example, the navigation tools for a program delivered on a floppy diskette with no sound or video can only afford (in terms of disk space) a static screen with icons and brief text explanation. This was the approach used in the X-Plore disk series.

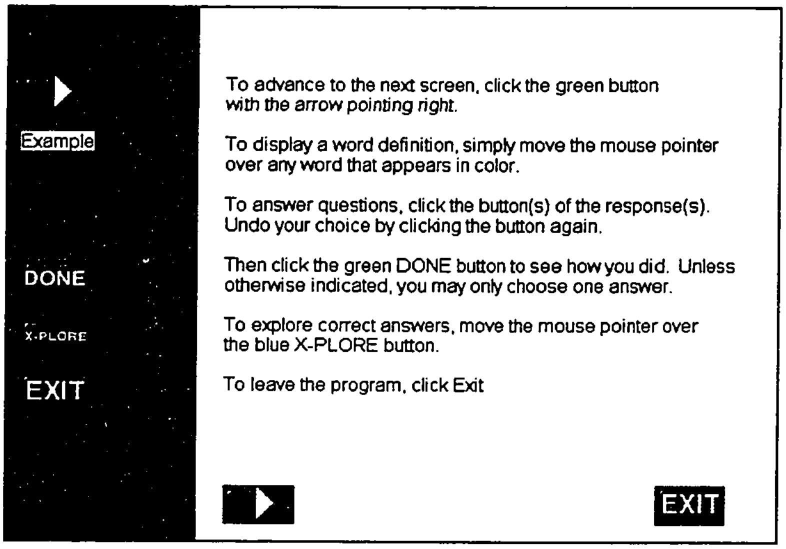


Figure 33: X-Plore single-page navigation instructions

In situations where the navigation tools may not be obvious (intuitive) to the users, or the interface is more complex, the interface may benefit from navigation instructions that use QT, AVI or Director movies demonstrating the actual use of (and results of) the available navigation tools. The Pacing Primer employed a navigational tour that introduced users to the program structure, as well as the instructional and navigational tools available to them.

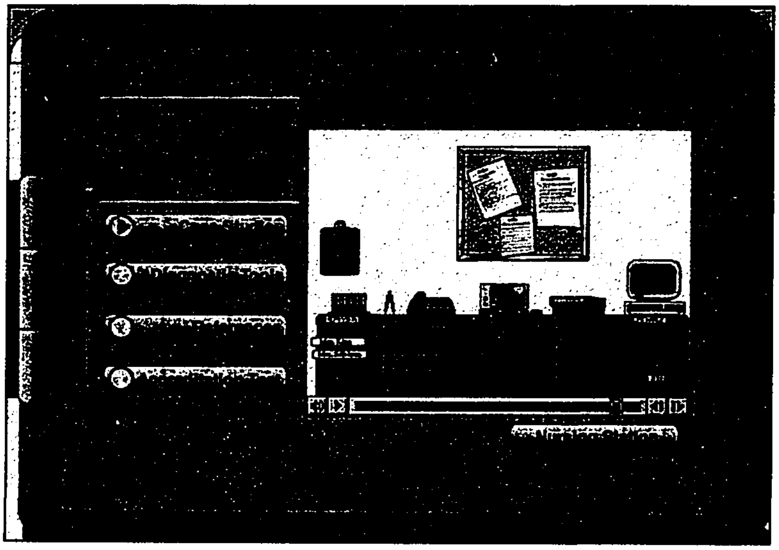


Figure 34: Pacing primer QuickTime navigation instructions

Accessing Help and Navigation Instructions

The universal protocol for help in most interfaces in the western culture is a question mark icon. The help menu should be accessible from anywhere in the program, at any time. Therefore, it must reside in a main menu area or be integrated as a reoccurring icon in all screens.

The recommended protocol for access or exposure to navigational tours is:

- The first time users sign into the program, they advance to the navigation instruction area and are immediately exposed to the navigational tools.
- At any time during the navigational tour (a QT, AVI or Director movie), users can stop, review or advance immediately to the 'main menu'.
- Users can always access the navigation instructions, at any time, from anywhere within the interface through the help icon.

The users' familiarity with an interface can be proportional to use, therefore the need for navigational assistance is inversely proportional. In a theory of display-based problem solving, Larkin (1989) proposed that short-term working memory represents task information against which production rules are matched. According to Lansdale and Ormerod (1994), this suggests that as they become more familiar with an interface, users acquire 'strategic knowledge' which improves their use of the interface itself. This supports the protocol that navigation instructions should decrease in detail each time a user logs on, or uses them.

As a result, three additional conditions to adjust the navigational instructions to the users' need for help can be added to the access protocol.

- The second time a user 'signs in', simple navigation instructions can be bypassed, advancing the user to the main menu.
- Complex navigation instructions should be reduced to a simpler set of instructions (e.g. text only) the second time a user 'signs in', and then bypassed the third and subsequent times.
- The interface should monitor the number of times navigation instructions are accessed. Exposure to subsequent navigation instructions or 'tours' should be adjusted accordingly the next time that user 'signs in'.

If the navigation instructions exceed more than one screen (or 30 seconds if it is a QT/AVI/Director movie) it is recommended that the instructions be indexed so that users can quickly select and review instructions on a specific subject, and return to the program in as short a delay as possible.

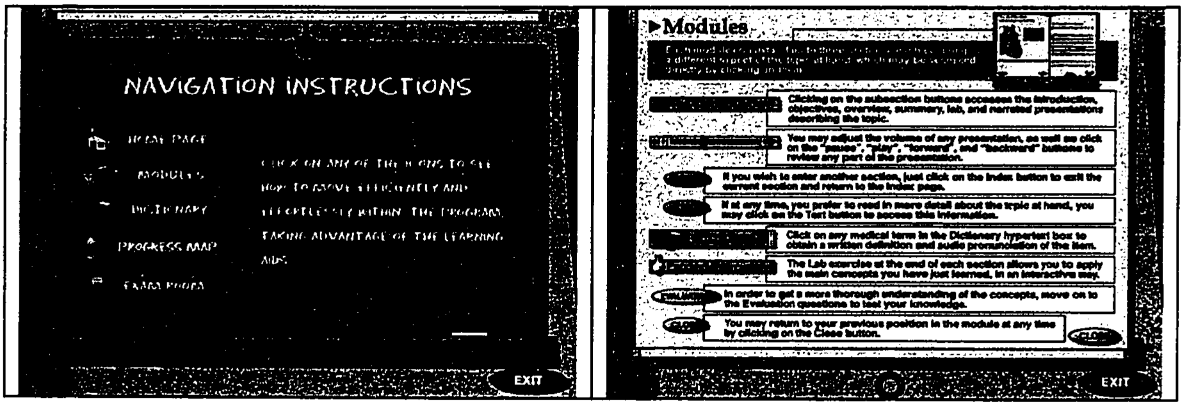


Figure 35: Backgrounder menu-driven navigation instructions

An example of reducing navigation instructions can be found in both the Backgrounder and Interactive Pacing Primer Interface. Once the navigational instructions or tour had been introduced and reviewed, it did not appear the next time that user logged-on. However, the navigation menu and tour were always accessible through the help icon.

Return, Enter, OK, Close and Exit

One recurring problem with navigational tools such as 'Return', 'Enter', 'OK', 'Close', and 'Exit', is a lack of standardization. In the interfaces discussed here, usage standards were established and applied.

During the development of the X-Plore interface there was some discussion as to the usefulness of the concept of 'Return' which can be problematic because it serves the same function as 'Back' (return to the previous screen). It can also be confused with the old standard of a carriage return, which with the modern keyboard, is closer to the concept of 'Enter'.

‘Enter’, on the other hand, is more of an operational tool than a navigational tool, best assigned as a device to complete an operation such as a mathematical calculation, or as a signal of confirmation. ‘Enter’ is usually synonymous with ‘OK’.

‘Exit’ is often misused as well when employed as a command to exit a screen, menu or physical layout area, which should be a ‘Back’ or ‘Return’ function. ‘Exit’ should only be used to lead towards the screen sequence which exits the program back to the host operating system.

‘Close’ is a definitive action used primarily for withdrawal from a single screen branch such as a pop-up dialogue box or simple help screen.

Forward and Back

In the interfaces discussed here, it was assumed that anyone using a modern computer capable of multimedia is familiar with a common VCR.

‘Forward’ or ‘Back’ may have many associated symbols, but its action should be standardized for use when moving along any linear sequence of events, or moving from one level to the next. ‘Forward’ is also interchangeable with ‘Next’. Although ‘Forward’ is a closer antonym to ‘Back’, it has more characters. As a result, ‘Next’ is often used when labeling an icon, or placing the word inside a right-handed arrow. Designers should be cautioned that the use of a right-handed arrow to indicate ‘Forward’ or ‘Next’ is only valid in Western culture.

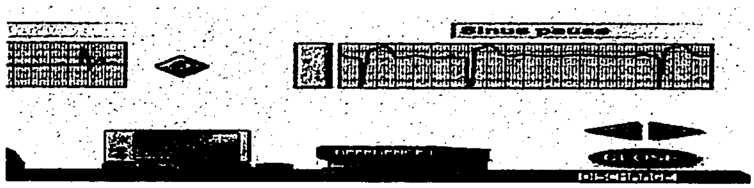


Figure 36: Standard use of 'Forward', 'Back', and 'Close' buttons

Note: The 'Back' command can also function as a 'Return' command when returning from a branch in a decision tree. However, the composite is not true for 'Forward', which cannot be used to advance at a multi-choice branch point.

QuickTime, AVI and Director Movies

QuickTime, AVI and Director runtime movies are navigation elements that mimic the common VCR, therefore the need for navigational instructions or labeling of these functions is questionable. Lynch and Horton (1997) observed that this type of interface also serves as both a control and a status indicator. Some interfaces remove the controls altogether, limiting users to one-pass viewing. This may be appropriate in situations where content retention is being measured or tested against time.

Scrolling

A standard in most web page designs, scrolling has the advantage of loading more material onto one page than can be seen in the defined window area. The designer must decide which is the most appropriate method for navigating large blocks of information: one long scrollable page or a series of sequential or branched pages.

Despite the disadvantage of adding to the 'thickness' of the program structure, sequential or branched screens have advantages. Pages -- and the information on them – are best presented in short, digestible bits to improve retention. Also, the visual analogy for a long or wide scrollable area is harder to design and see, since only a fraction of it can be displayed at any one time.

The need for a scrollable page usually indicates too much information for one screen, and should prompt a reevaluation of the content structure. However, there are situations when a vertical and/or horizontal scrollable page may be appropriate, such as with large maps, photographs or graphics that should not be segmented, yet cannot be included on one page/screen because of the density of information, size of labeling text, etc. Long blocks of text or lists such as those found in searchable lexicons would also merit a scrolling function.

Pop-ups or Dialogue Boxes

Familiar in all commercial software, pop-ups/dialogue boxes are reversible sub-routines used for single-level navigation (e.g. hypertext), or confirmation.

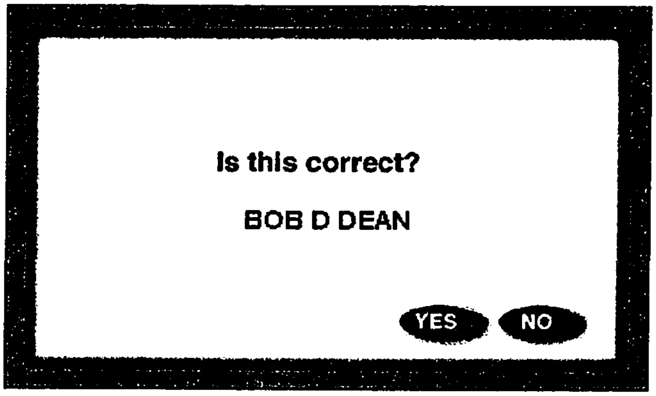


Figure 37: Confirmation dialogue box

Standardized Tools

One easy option for navigation tools is to use standard protocols by emulating those used in the Microsoft or Mac OS operating environment. A key deciding factor is the audience and scope of the content. The use of standard MS or Mac OS icons and menu tools capitalizes on widespread familiarity with existing navigational functions. However, the majority of interface tools are not applicable in a limited-use environment, (e.g. there is no need for the filing, editing, viewing, inserting, formatting, tools, tables or window function).

The navigational tools used in the X-Plore and Backgrounder series and the Pacing Primer avoided the standards used in commercial operating systems in favor of 'light' navigation with very few tools, and very few levels behind the tools. In an interface intended for limited use, users will need only a fraction of the usual number of interface tools. By design, fewer navigational tools means the interface is easier to learn and use because it is closer to the users.

Note: A 'light' interface should not be confused with the concept of an 'intuitive' interface which is a function of recognition memory. The interface design should take advantage of both as often as possible.

Icons

According to Meggs (1989) and Norman (1990), graphical user interfaces (GUI) are effective because they exploit graphic metaphors such as symbols, pictograms, icons, etc. Intuitive interface semiotics exploit memory recognition with objects or symbols that are familiar to the majority of users in western culture. In what Larkin (1989) refers to as display-based problem solving, users can exploit externally available information (self explanatory icon or symbol) which reduces the load on short term memory. Or as Lansdale and Ormerod (1994) observe, it increases the emphasis on design, and decreases the emphasis on learning.

Examples of these principles are seen in interface icons where:

- A door or exit sign indicates that users can leave the program.
- A book marked 'Dictionary' indicates that users will access definitions.
- A printer indicates that a print-out will occur.
- A loudspeaker indicates that users will hear something.
- A movie projector or film strip indicates that users will see a film/video sequence.

With some subjects and/or audiences there exist specialized icons, objects or symbols which are easily recognized by users, but not by a general audience. For example, in the Interactive Pacing Primer:

- A file folder with colour tabs is the universal filing method in the medical environment. Users know it will contain historical information on a patient.
- Calipers (similar to dividers) are a familiar assessment tool for anyone working in electrophysiology, and are associated with ECG interpretation.
- The 'programmer' is a machine unique to pacemaker installation and maintenance. Users know it will produce useful data from the interrogation of the patient's pacemaker.

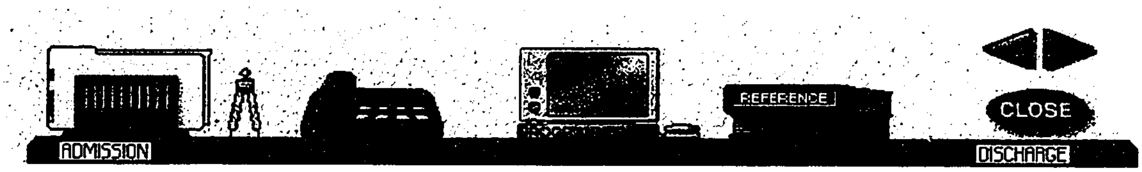


Figure 38: Icons at 'Nursing Station' in Pacing Primer

The navigational tools used in the Interactive Pacing Primer interface are an example of intuitive semiotics. Although there are many navigational icons with several levels behind them, the navigational tools are all familiar to the users, and the underlying structure mimics normal clinical procedures.

'Specialized' icons must still adhere to Lansdale and Ormerod's (1994) four characteristics for any object in an object oriented system:

1. Objects should contain physical or conceptual entities that exist within a problem domain.
2. Each object should have an action associated with it.
3. All objects should only be activated via user input.
4. All objects should be grouped and classified by common attributes.

Tip: Validating icons

If the designer is not familiar with specialized icons, objects and symbols, it is advisable to run a small validation with a sample of the target audience. This will determine if the icons, objects and symbols are clearly and quickly understood, and whether or not they have meanings other than the ones intended.

A quick validation method for specialized icons:

1. Mix all of the specialized symbols being considered with at least 25% distracters and 25% universal icons.
2. Print them out on a sheet with a space next to them.
3. Distribute them to a sample of the target audience asking them to:
   - describe the symbol in as few words as possible;
   - describe what they would expect to find if they clicked on it;
   - if they guess, a question mark should be added after their description; and
   - if they do not understand the symbol at all, note that in the description.

Orientation: The Content Compass

At all times users should know where they are within the program (location), and where they are in relationship to material covered vs. material left to cover. Added to the four structure related issues offered by Baecker and Small (1990) and Cole, Lansdale and Christie (1985), users should always be able to answer these six questions:

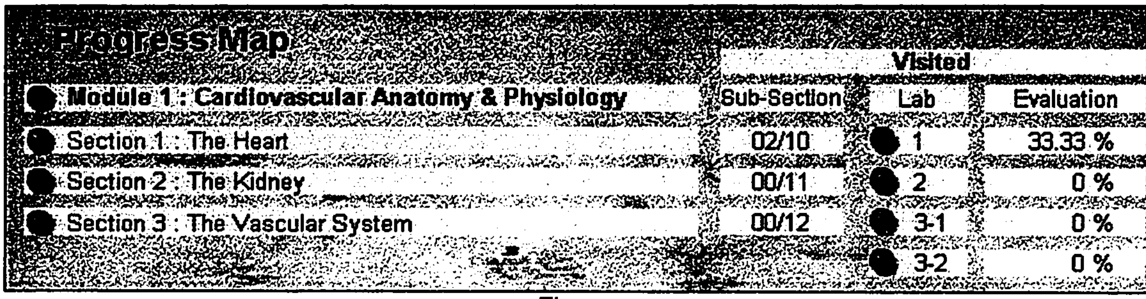
1. Where am I?
2. How did I get here?
3. What can I do here?
4. Where can I go next?
5. How much have I done
6. How much more do I have to do?

For structures that are manageable within one screen, Lynch and Horton (1997) recommend hierarchical diagrams because they are familiar to most users who can easily adopt metaphors built on mental models. Since not all users are spatially oriented, a structural map be helpful only to certain users.

Traditional methods are also employed to provide users with on-going feedback on their progress through sections of the interface. These include hierarchy indicators such as chapter, section and page numbers. In the X-Plore series, the structure was simple enough for each screen to indicate progress with only two numbers: one indicating questions covered; and the other, the total number of questions in that group. Horizontal progressive bar graphs are adequate within linear sequences where users are forced through all the material. But in an interface where not all material is mandatory, linear devices are not appropriate.

As a result, a simpler, single screen navigational tool that represents location, progress, and performance within a complex structure was developed for the Backgrounder series. One single-screen progress chart represented the content structure, and by changing colour or with highlighting, indicated to users:

- how much material they had covered, and had yet to cover;
- a global picture of where they were located within the content structure; and
- an on-going report on their performance from any evaluation mechanisms they had completed up to that point.



Figure

Figure 39: Segment of orientation screen from Backgrounder series

'Wait' indicators (e.g. hourglass and watch) are standard to certain interfaces and operating systems. 'Wait' states are inconvenient, discouraging, and should be reconsidered. They are, in effect, a planned inadequacy that increases the memory load of users in order to cover basic interface and instructional design flaws. If movement within the interface requires 'waits', then perhaps too much information is being loaded. Whether it is a text, graphic, photograph, or video clip, the length and resolution of the incoming data should be reevaluated, and the programming approach to I/O perhaps rethought as well.

Confirmation

There are two polarities for confirmation in navigation: positive confirmation that provides reinforcement and encouragement (and should be used frequently wherever appropriate), and negative confirmation, (generally avoided in the interfaces discussed here). Users should not receive negative feedback from an interface, except for corrective responses in quizzes or tests.

Users need instant and positive confirmation that input has been detected. I/O delays may mislead users into thinking that the device is not active, or not reacting to their input. And when selecting an object within the interface that is not 'Hot', the absence of a reaction is the only feedback available. This is not a problem for command input (typing) because characters appear on the screen, nor with pull-down/out menu's where a dialog box opens. However in the case of a clickable selection, visual or auditory confirmation should be provided.

Visual Cues

Visual cues are easier to design and program, and do not suffer from the problems that can plague audio cues. To provide noticeable visual cues:

- slightly offset, then return the icon or object to its original position;
- highlight the object;
- reverse the shading;
- change the object's colour; or
- have the object make a natural move associated with its function.

An example of the latter is found in the Interactive Pacing Primer main menu where -- when selected -- the cover of the book opened, the handset of the phone lifted, the calipers opened wider, etc.

Although it is generally agreed that the more explicit the feedback, the better the comprehension of its function, there are limits to the required depth and resolution of confirmation. Icon reactions in all three interfaces discussed here were limited in detail to one or two frame variances.

Audio Cues

There are many reasons for using various audio techniques as confirmation mechanisms to support visual cues. Those listed by Lansdale and Ormerod (1994) include:

- mixing iconographic representations with acoustic attributes improves recognition of function and outcome;
- interfaces for the visually handicapped (beyond the scope of this discussion);
- reduced visual processing through parallel input by spreading the information load across more than one sensory input (i.e. audio/visual); and
- auditory inputs using natural sounds associated with the object that speed-up or reinforce its identification.

Audio cues are simple and effective, assuming the platform has audio capabilities. However, audio cues can quickly become annoying if there are many such confirmations, as one would find in a 'fast' interface[9]. One solution to the annoyance factor is a functional icon which enables users to turn the audio off. But this creates three potential problems:

1. It thickens the interface with another tool which must always be available.
2. Additional navigation instructions are required to explain the tool.
3. By turning off the audio, users may be turning off other important audio outputs such as narration, audio portions of video, sounds associated with process. (One solution to this problem is an icon which only turns off the confirmation audio cues. However this adds another navigational instruction.)

In addition to the quality and depth of confirmation, Lansdale and Ormerod (1994) also evaluate how 'user-centered' a design is by measuring how much user control it offers in terms of predictability and reversibility. While predictability is simply a function of consistency, the designer should take into consideration when and where reversibility is appropriate. In situations such as a quiz, reversibility of any navigation function may not be desirable because it decreases the accuracy of quiz results. In most other navigation structures, reversible movement through the structure is preferable to 'jumping' from one branch to another which can disorient or lose the users.

[9] A 'fast' interface is one in which the user moves about quickly, executing many decisions in a short period of time... such as when traveling back or forward through the structure without having to stop to perform a function each time.

Transitions

As Lansdale and Ormerod (1994) note, travel within an interface is abstract, so to avoid disorienting users, transitions from one 'state' to another must represent an action that is acceptable in real-life. Transitions fulfill this purpose.

Examples include the Pacing Primer navigational 'tours' that are imbedded in a 'help' clipboard that hangs on the wall, or the lowering and raising of the desk to signify moving from the main menu to a particular instructional mode, and at the same time providing more room for the navigational tools.

Important to transitions is how the host processor and graphic card handle the change in screen dynamics. Key screens between which users must move should be simple, with low colour depth and small file volumes in order to make transitions as smooth and as natural as intended. If the processor and card can't handle the transition, it may be quite disturbing. Therefore:

- Keep key screens simple with low file volumes.
- Keep transitions simple with few 'in-betweens.'
- Keep the number of key screens and menus to a minimum.
- Test all transitions before committing to them.
- Test all transitions on a variety of the target install-base.

Print Navigation Instructions

Many interactive programs are distributed with instructions. For the type of programs demonstrated in this discussion (limited-use), it is recommended that navigation instructions not be included in print material.

- If users require a manual to learn how to navigate the interface, the interface is too complex and should be rethought.
- Instruction booklets easily become separated from diskettes or CDs, potentially leaving users with no instructions.
- Experimentation is the most common form of learning an interface (Lansdale & Ormerod, 1994). Encouraging users to use print navigation material discourages them from becoming familiar with the interface.

Ideally, the only print instructions should be the brief install routine, printed on the diskette label or disc carrier.

Step 8: Evaluation, Feedback, Tracking and Reporting

One area of interactivity that distinguishes CBT from traditional instructional methodology is the ability of the program (and the program designers) to:

- evaluate users' performance;
- provide feedback to users on their performance;
- track users' progress and interaction within the program; and
- report users' performance and opinions back to the designers;

Most of these functions are imbedded within the interface, transparent to the users. At the same time they provide the interface designer and end-client with valuable information that can be used to improve their respective performances.

Evaluation of User Performance

There are various interests and objectives to be served by evaluation mechanisms in an interface. Clients want to know that certain critical concepts have been acquired and retained while specific levels of proficiency have been achieved. In addition, clients require evidence that the significant resources that go into each training initiative provide a return on their investment, supported by measurable results delivered in as short a time delay as possible. Conversely, designers want to collect data that will confirm the effectiveness of various interface design elements, and where there is room for improvement.

In the X-Plore series, users were given feedback on their performance (e.g. "you have correctly answered 4 out of 5 questions") before being referred to the remedial loop. However, the performance data reported to the developers and client was somewhat different. It included a score based on the number of questions answered correctly before entering the remedial loop, time spent on the system ('time-on'), and users' perceived prescribing preference (post module exercises that did not count towards the final score).

Conversely, in the Backgrounder series, user were given feedback on quiz results which appeared at the end of each lab, and 'Exam' results which appeared at the end of each module. But only the score from the final exam was reported to the data collection center for use by the developers and client.

In the Pacing Primer, users were given the score of their final exam upon completion. This score was reported directly to the client so that they could issue credits for self-guided interactive learning offered by professional associations and governing bodies. These continuing education (CE) credits are required by state/provincial and federal regulating authorities for the renewal of licenses of healthcare professionals. In order for the Pacing Primer to qualify for CE credits, the interactive programs had to record and provide measurable evidence that learning had taken place

User Feedback Mechanisms

There are several concepts or levels embraced in the educator's definition of feedback, each measured by its complexity and value to the users. Performance feedback can be classified by seven levels.

1. No immediate feedback whatsoever (score only upon completion).
2. A mark (score) only.
3. Indication of which questions were correctly and/or incorrectly answered.
4. Indication of which selection(s) in a question is/are correct and incorrect.
5. Explanation of why each selections is/not the correct choice.
6. Referral to the area of content where the correct response can be found.
7. Option of a remedial question.

Level 1 Feedback

The most familiar is the traditional exam-now, mark-later where the score is given as early as at the end of the quiz/test/exam, or at a much later date by some central reporting mechanism. Performance feedback on a delay is not ideal, but is necessary in some situations whether intentional (score at end of test rather than on a per question basis), or due to restrictions of the evaluation mechanism (traditional grading systems).

Level 2 Feedback

The simplest form of immediate feedback is a green check mark or a red 'X' next to a response. If the answer is wrong, users do not know what the correct selection(s) is/are. This is necessary in situations where users may be required to answer the same question again at a later point.

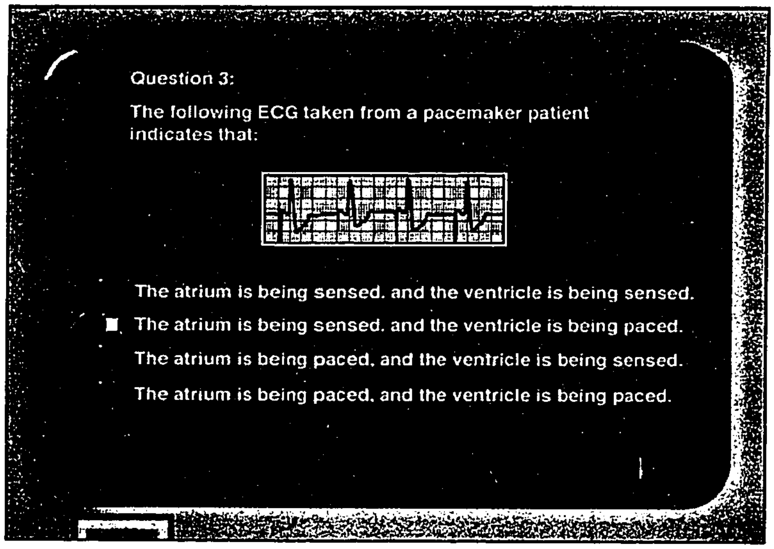


Figure 40: Level Two feedback in the Pacing Primer

Level 3 Feedback

The next level increases the value of the feedback with the addition of a green arrow to identify the correct response. By providing a more complete form of feedback, it also makes it easier for users to correctly answer the question again at a later point. This is what prompted the design of the 'question conversion' remedial loop in the X-Plore series.

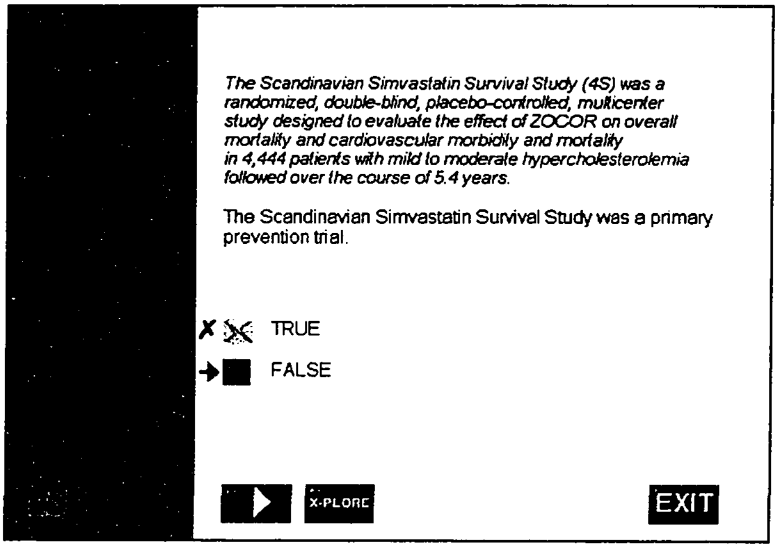


Figure 41: Level Three feedback in X-Plore series

Level 4 Feedback

In the X-Plore series, the addition of an 'X-Plore' button next to each answer opened a pop-up window that provided the rationale as to why or why not each particular response was correct or incorrect. The complexity of such a function can run the range from simple text to full AV support.

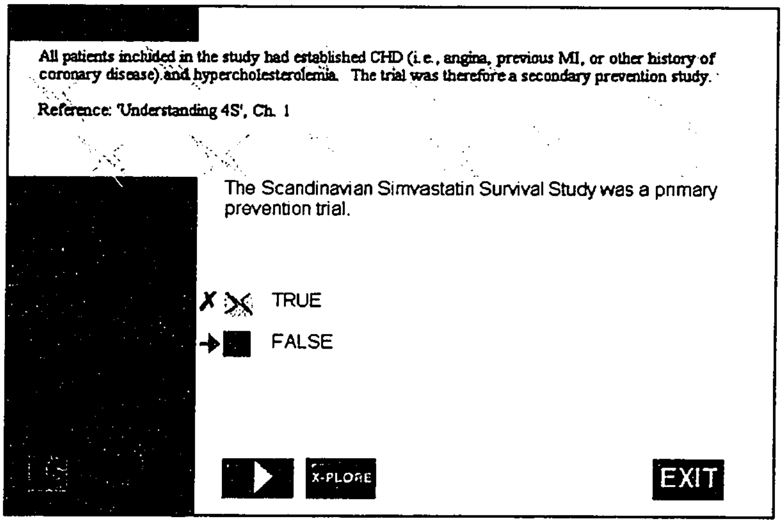


Figure 42: Level Four feedback in the X-Plore series

Level 5 Feedback

Directing the learner to the appropriate remedial resources opens a broader feedback base, depending on the extent of the remedial resources and how directed the referral is (i.e. should it direct users to a paragraph in a text, or an entire chapter in a book?) In a more interactive system, the program itself channels users into remedial interactive exercises.

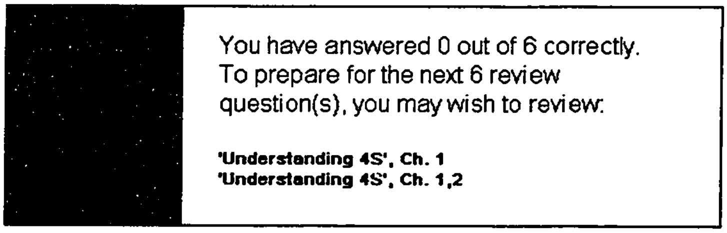


Figure 43: Level Five feedback in the X-Plore series

Level 6 Feedback

By providing a remedial question, Level 6 feedback re-tests users' knowledge acquisition, and the effectiveness of the remedial loop.

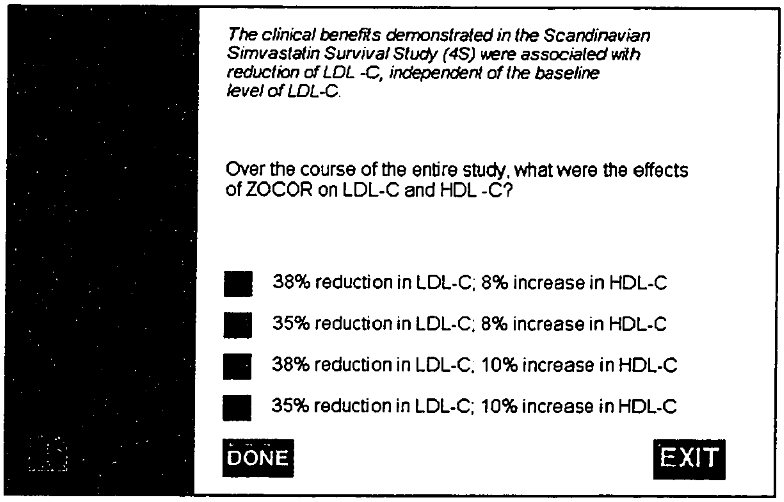


Figure 44: Level Six feedback in the X-Plore series

Level 7 Feedback

One final feedback mechanism is a graphical summary score page that displays a specific user's performance in all areas of the program, rather than a global score. This mechanism provides a higher resolution of feedback and passes the control of remedial activity to the users.

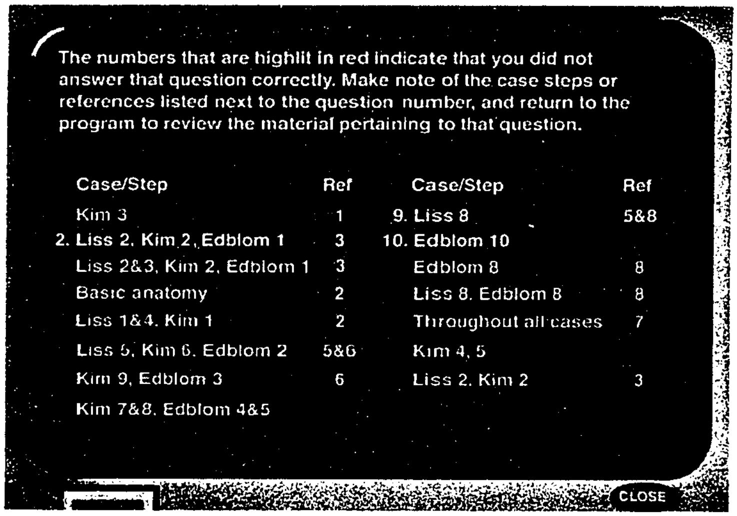


Figure 45: Level Seven feedback in the Pacing Primer

Tracking Mechanisms

Aligned with the appropriate reporting mechanism, tracking can provide useful information beyond performance scores. Tracking can also provide data on:

- How many remedial cycles were required before a question is correctly answered. This information has to be weighted with the type of feedback and remedial information that users are provided between attempts.
- How many remedial levels did users have to go through in order to correctly answer a question.
- How long users interacted with the program from entry to completion. This measurement requires a keystroke-per-minute evaluation mechanism that differentiates between user time-on and user time-out.
- How long it took users to answer all questions in an evaluation segment. Converting time-on duration recorded during an evaluation procedure to a meaningful quantitative value is also a complex procedure subject to many variables such as:
  - is the performance being evaluated time-sensitive?
  - were users actively interfacing with the program during the measurement (or on the phone?).
- Whether or not users accessed adjunct resources, or remedial content after incorrectly responding to a question.

The author is interested in exploring this area of evaluation in future studies.

Reporting Mechanisms

Tracking mechanisms must be concurrently designed to coordinate with the reporting system. Since most of the interactive training programs discussed here were delivered to the field by courier or direct mail, a methodology for collecting performance data beyond the alpha/beta-test stage was required. The result was a flexible performance reporting tool with the acronym LVIS: Learning Validation Information System.

With the LVIS, once a user had completed the required evaluation components of a program, tracking information was converted to an eight to 16 digit algorithm which included the user's employee ID number or name. The tracking operated in the background, transparent to the users.

A variety of reporting methods can be used, depending on the type of information being reported, and the client's resources and preferences. Specifically, LVIS incorporates several methods for collecting data from the field:

1. Business Reply Card
2. 1-800 touch-tone
3. 1-800 voice recording
4. Auto-modem transfer
5. Background e-mail

Business Reply Card

A 16 digit[10] code generated at the end of the program is manually transcribed onto a postage pre-paid Business Reply Card (BRC) that is included with the program. The card also contains program evaluation questions, and one area for write-in responses. Upon completion, the program directs users to transfer their code, complete the feedback questions, and drop it in the mail. BRCs are easy, simple and inexpensive... a great 'first step' for first-time clients that want to cautiously venture into evaluation of interactive delivery.

Limitations include:

- A limit to the amount of data that can be tracked and transferred because of algorithm limitations.
- Users can make errors when transcribing the code from the screen onto the card.
- Data entry clerks can make errors when transcribing the code from the card into the database. (Erroneous codes are rejected by a simple sum-check sub-routine in the evaluation database).
- It is labor intensive for data entry clerks to transcribe the codes from the card into the evaluation database. (Research into optical character reading (OCR) devices revealed that they were too expensive for limited applications).
- Users may not always mail their cards due to forgetfulness, or may lose the card. (E-mails are sent to user's reminding them to return their BRCs).

The X-Plore series has used the BRC method for program evaluation with satisfactory results with close to 100% returns.

[10] 4 digits in groups of 4 =16... the average amount of digits on a credit card

1-800 Touch-Tone

This method establishes a 1-800 number with simple voice interactive instructions to transfer the code using a touch-tone phone.

Advantages:

- More user-friendly, resulting in a high percentage of returns.
- Eliminates erroneous codes. Host system does an immediate sum-check and advises users at the time of entry if their code is invalid.
- Eliminates labor and errors common with manual data entry into a database.

Disadvantages:

- Write-in (qualitative) responses are eliminated (although multiple choice program evaluation questions can still be handled by the code).
- The cost of setting up the 1-800 system which includes:
  - a dedicated receiving PC and software;
  - several modems;
  - a 1-800 number; and
  - a series of ring-down lines (necessary if codes arrive in a short time period).

To date, no clients have requested this reporting option.

1-800 Voice Recording

This method uses a voice mailbox to which users dictate their exit codes, and any other information. The telephone number, type of information and sequence is prompted by a reporting screen at the end of the program.

Advantages:

- Eliminates the mailing component, resulting in a greater percentage of returns and more immediate results.
- Allows for more detailed and greater amounts of information to be transferred and recorded such as name, address, and comments.
- Relatively inexpensive, requiring only the cost of an 800 number, and the dedication of one voice mailbox.

Disadvantages:

- Users can make errors when dictating the code from their screen.
- Labor intensive for data entry clerks to transcribe the information from voice mail into the evaluation database.
- Data entry clerks can make errors when transcribing the code from voice mail into the database;
- The cost of the 1-800 number.

Because the client needed personal data on each user in order to issue Continuing Education credits, this reporting option was the natural choice for the Pacing Primer program.

Auto-modem transfer:

This methodology is similar to that used for the automatic registration of commercial software at the time of installation. Users initiate the code transfer by clicking a box that says "send results now". The system's resident modem dials up a dedicated computer which receives and stores the data.

Advantages:

- Information recorded by the program's tracking mechanism is not limited to one coded algorithm. More detailed tracking can be executed and transferred to the host evaluation system.
- More qualitative feedback can be solicited from users in the form of type-in program evaluation response forms.

Disadvantages:

- Users' computers must have properly configured and functioning modems.
- The cost of setting up the collection system which includes:
  - a dedicated receiving computer and software;
  - several modems;
  - a 1-800 number; and
  - a series of ring-down lines (necessary if many codes are expected in a short time period).

To date, no clients have requested this reporting option.

Background e-mail:

With background e-mail, once users have completed the required components of the program, their computer automatically generates and stores the exit code. The next time the user logs onto the company's e-mail system, their computer transmits the code (and any other tracking data) via e-mail to a dedicated data collection file.

Advantages:

- Requires no conscious input from the users.
- Accurate reporting, no room for human error.
- 100% return (except for users who do not complete the program).

Disadvantage:

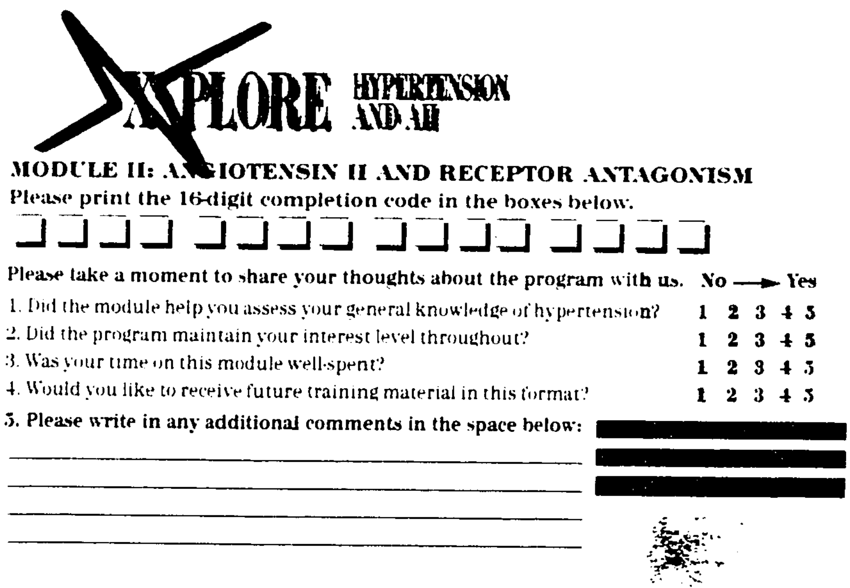
- The user-base must have regular access to a common e-mail system.

The Backgrounder series used this reporting option for three complete series covering 14 modules. This is a preferred method for the collection of evaluation data for interactive training distributed within a corporation.

Tracking and reporting in the X-Plore series

In the X-Plore series, the number of attempts to achieve 100% was reported on a per module basis. This number was compared with the total number of questions and embedded in a 16-digit completion code that also contained their territory code (employee PIN number). This code was generated at the completion of every second module and returned to a central data collection point via a postage pre-paid Business Reply Card (BRC) where the completion codes were entered into a database, and reports generated.

The X-Plore series, BRC cards also contained space for qualitative evaluation using four questions with a scalar response and one write-in comment area.



XPLORE

MODULE II: ANGIOTENSIN II AND RECEPTOR ANTAGONISM

Please print the 16-digit completion code in the boxes below.

Please take a moment to share your thoughts about the program with us. No → Yes

1. Did the module help you assess your general knowledge of hypertension? 1 2 3 4 5
2. Did the program maintain your interest level throughout? 1 2 3 4 5
3. Was your time on this module well-spent? 1 2 3 4 5
4. Would you like to receive future training material in this format? 1 2 3 4 5
5. Please write in any additional comments in the space below: [redacted]

Figure 46: BRC from X-Plore series

Tracking and reporting in the Backgrounder series

In the Backgrounder series, the results of the 'Final Exam', along with responses to seven program evaluation questions, were imbedded in a code generated upon completion of all modules in the series and the 'Final Exam'. This code was transmitted to a central data collection point via 'background e-mail', whenever users logged into the companies e-mail system. The codes were then decoded, entered into a database, and reports generated.

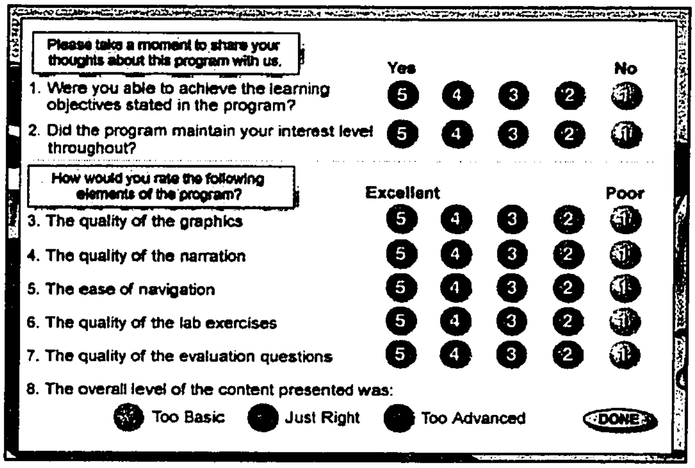


Figure 47: Evaluation screen from the Backgrounder series

Tracking and Reporting in the Pacing Primer

Tracking in the Pacing Primer was limited to the final 'Test' score and a four-question evaluation screen. The data was converted into a six digit completion code and submitted using a 1-800 voice mail reporting system. At the time of reporting, the reporting screen of the program prompted users for the additional personal data that was required in order to issue the appropriate continuing education credits.

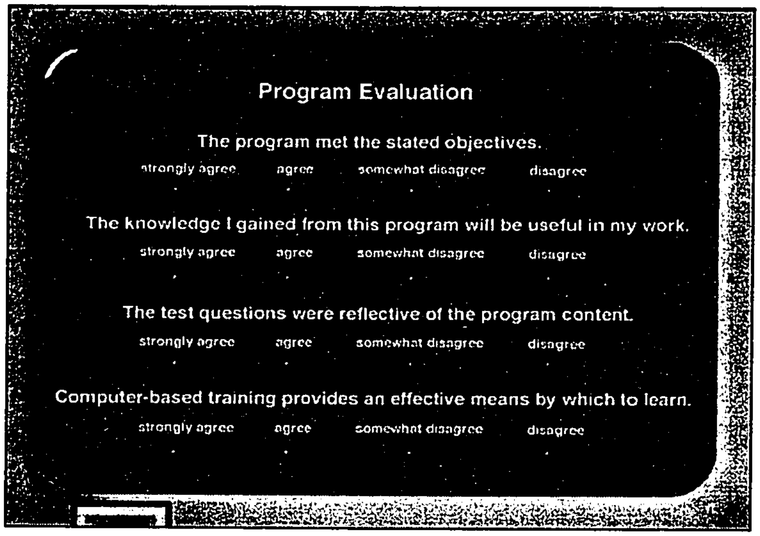


Figure 48: Evaluation screen from the Pacing Primer

Step 9: Approval and Evaluation

Once an interface is constructed, tested and perfected, the developer usually tries to amortize the investment over a series of programs, using the interface as a template. This was the case for the X-Plore and Backgrounder series. Whereas content can be easily added, removed or modified, the same is not true for the interface itself.

Because of the complexity of an interactive interface, it is important to get the client's approval on all material before it enters the programming stage. There is a great liability in going into production without an approved interface. Corrections at any point after production has started are time consuming, costly, and run the risk of pushing a program over budget, over schedule, or both.

In order to get approval on program content, it is important to present it in context (wherever possible) to clients so they can see the interface on which the material is being delivered. This process of approval starts during the program outline and submission stage where samples of the interface are demonstrated with small portions of content.

There are two elements of approval, one involving the interface, the other involving all content prior to programming.

Approval of the Interface

Time and resources permitting, the interface should pass through two approval procedures prior to programming: once 'on paper' and once as a working demo on diskette or disc.

Paper Interface Approval

The most common method of pre-programming approval is to present visual samples or sketches of the key interface screens, along with a flow-chart that indicates how these screens are linked within the navigational structure. This method is quick and inexpensive, but does carry some limitations:

- The people approving the proposed design may not 'see' the interface from sketches and structural charts.
- The actual dynamics (action over time) of the interface are difficult to assess.
- Qualitative interface attributes (e.g 'feel' or 'character') are difficult to assess.
- Nielsen (1992) found that some usability problems (e.g. missing interface elements) were more difficult to find through heuristic evaluation using 'paper' prototypes.

Demo Approval

A more effective option is to construct a demo version of the interface, populate it with a few samples of instructional and navigational tools, and test it with both the client and a sampling of the target audience. This is a more manageable variation on the detailed and extensive 'cognitive walk-through' approach of Lewis, Polson, Wharton and Uyeda (1990). This method is expensive, consumes a lot of time and resources, and therefore not appropriate for all projects.

However, given the overall cost of developing an interactive training program -- especially a series -- the investment made in qualifying the interface before it goes into production will result in greater customer and user satisfaction with the interface, lower probability of modifications to the interface during the production period, and a better return on investment.

Designing a Demo

It is advisable to field test a working example of the proposed design. But, the process of constructing an interface, developing a 'mock-up', running samples on users, and evaluating the results is time consuming and expensive.

In the case of the Interactive Pacing Primer, the interface design was evaluated by 56 people representing the target audience during an annual association conference in Anaheim, CA. The objective behind the demo was to enable the target audience to 'interact' with as many of the interface tools as possible, in order to provide meaningful feedback.

The challenge in creating the demo version of the interface was to create an affordable yet workable facsimile that did not require a user learning curve that exceeded the demo interaction time. Users have to invest a certain amount of time in order to become comfortable with any interface. However, a demo version can only offer a brief sample of the content. Imagine the frustration of users who spend five minutes learning an interface, in order to spend only five minutes of interaction time with the program content.

The solution was a six minute QuickTime free running movie that demonstrated the interface and its principal tools. Few user input prompts were integrated at the beginning (when users are most unfamiliar with the interface) and more at the end (as users become more familiar) to measure the users' ability to 'learn' the navigational tools in a short period of time.

Testing a Demo

Conference attendees who expressed interest in the demo version (displayed at the client's booth on the trade show floor) were invited to a suite where they could 'test drive' the interface on one of two test stations. The entire demo cycle was designed to last only 15 minutes, processing eight people an hour for four hours a day over two days.

The demo cycle included:

Activity	Time
Brief explanation of purpose of demo	1 minutes
Brief introduction to basic navigation (mouse)	1 minute
User navigates demo	6 minutes
Completion of evaluation form	3 minutes
Personal debriefing	3 minutes

Table 16: Demonstration cycle for testing demo interface

Because of the positive response to the demo, debriefings often lasted five to ten minutes. As a result, each cycle was approx. 1/2 hour instead of the planned 15 minutes. 56 questionnaires were completed. The programmer and a member of the interface design team conducted the debriefings.

This exercise provided sufficient feedback on the interface to confirm which components were effective, and prompted several changes.

- Content handling functions such as 'Highlight', 'Copy', and 'Paste' were not perceived as useful. [11] Their removal simplified the principal interface tools.
- The audience responded very positively to ECG-based exercises. These became one of the principle instructional tools in the final interface.
- The audience was impressed by the animation of the cardiac conduction cycle and the use of patient interviews. This prompted a greater emphasis of these and several other instructional tools in the final interface.

Note: Data from the SMEs field training experience, along with FAQs from the clients 1-800- technical help line, were also used to shape content. It is not within the scope of this discussion to discuss the approval process for medical content.

Approval Stages During Production

The client approval process is an opportunity for the client to monitor and evaluate the designer's ability to convert the client's vision into an instructional vehicle, giving life to the interactive program. During production there are several approval methods used to obtain client sign-off as early as possible, and at the same time ensure that the client understands both the navigational and instructional context in which the content will be presented.

[11] A 'Print' function that was evaluated as useful was later removed because of the problem of including and updating printer drivers for an undefined install-base.

Approval of Interface Screens.

As soon as they are available, colour print samples of key interface screens should be sent to the client for review and approval. The client initials each approved page or marks-up required changes, and returns them to production.

The client should be asked to sign-off on:

- Interface visual metaphors (main menus)
- Colour scheme
- Text font and size
- Background textures
- Icons
- Graphic standards (style, size, position)
- Navigation tools
- Use of corporate logos, etc.

Because of the simplicity of the interface of the X-Plore series (which was predominantly text), most approval discussions were limited to content. In the case of the Backgrounder series, the interface was an evolution of an earlier, successful interface, and a lengthy submission process at the conceptual stage. And with the Pacing Primer program, there were very few modifications at this stage, partially due to the client's familiarity with the 'demo' that had been developed and tested earlier.

On average, interface elements can be expected to go through at least two drafts prior to approval. However, this will vary depending on the client's experience with the process, and the clarity of their own vision of the final product.

Approval of Text Content

All material that appears in text on the screen or in audio and/or video segments should be submitted both in print and by e-file. The client should sign off on:

- On-screen text
- Narrative text
- Navigation instructions
- Intro/outro narrative
- Evaluation exercises Quizzes, case studies, text
- Resource text
- Audio segments (audio scripts)
- Video segments (video scripts describing what would appear in video segments)

All elements, visual or text, must be approved by the internal medical review mechanisms within the respective institution. This can be a lengthy procedure, and the production schedule should take this into account. However, changes made at this stage are relatively easy and inexpensive when compared to the consequences of modifications made once the content goes into production.

Approval of AV Elements

All graphics used in the program should be laid out in a story board format so that the client can see the synchronization of rough source graphics with associated text. The client should be asked to sign off on:

- The original graphic source
- A rough example of the graphic
- The position of the graphic within a screen
- The position of the graphic within a sequence

The same procedure applies for proposed stock animation or video sequences.

Samples of the proposed voices for both narrated and other audio sequences should be pre-selected and forwarded to the client for approval. This should also include music and sound effects if the design calls for them. Although clients are usually most concerned with the principal narrators, any character voices should be approved as well.

The client should also be asked to approve photographs of the actors cast for on-screen appearances or dramatizations, and when possible, after having viewed existing video samples of their work. Once voice narration has been recorded, and on-screen talent has been shot, edited, transferred, compressed and digitized, the replacement cost is prohibitive.

In the case of non-professional on-screen talent used for interviews, testimony or demonstration, (e.g. nurses or doctors), it is advisable to have the client select and approve them. Where patients are required, the doctors usually make arrangements with their own patients.

The client should also be supplied with a simple flow-chart of the program structure. Although the client cannot be expected to understand a structural diagram sufficiently to sign-off on it, it can be useful in helping the client understand where all of the material being produced is positioned within the structure.

For example, in the Pacing Primer, any time the client was asked to approve any program material, he/she knew:

- where that material went within the program;
- what it looked like (or a rough idea of what it would look like);
- where it appeared on the screen;
- which text appeared concurrently on the screen, and where;
- which graphic, audio or video segments would be included; and
- what the sources were (where relevant).

Production schedules in the real world

Regardless of how well a program is scheduled, there is always the reality that clients may have trouble meeting deadlines, either in terms of delivering source material, or approval at the various stages. Therefore in the face of collapsing production schedules due to approval delays, it is sometimes necessary to evaluate the risk of producing interface elements prior to approval. Knowing this, it is advisable to take preventative measures during the contracting period by defining and agreeing upon a mutual production schedule with specific:

- dates for submission of source material from the client;
- production house turn-around time;
- dates for submission of material for approval to the client;
- client approval turn-around time; and
- dates for final approval from the client.

If and when approval deadlines are missed, delivery dates are rarely adjusted accordingly. Therefore it is often necessary to begin production on interface elements prior to approval. Otherwise, deadlines will not be met once approval is received. Beginning production before approval has been received requires a certain amount of risk management.

Elements that can go into production before approval, if necessary, include:

- Elements supplied by or recommended by the client.
- Key interface elements such as main menus, intro and exit pages, backgrounds, etc. The client should see these as early as possible.
- Tools using non-original sources, or royalty-free sources such as clip-art icons, graphics or photographs. These can be changed at little cost.

Elements that should never go into production prior to approval include:

- Any video recording prior to approval. The cost of re-shooting, editing, compressing and digitizing can double video costs. In the case of interactive medical training programs, medical professionals such as doctors are rarely available to 'do it again'.
- Any audio recordings. Re-reads, re-editing, sampling and inserting often outweigh the initial time gained by proceeding prior to approval.
- 2D or 3D animation is time and dollar intensive. Even the simplest changes are time intensive in terms of rendering, and labour intensive.

Interface Evaluation

There are several recommended stages and methods of evaluating/testing interfaces prior to final release to the field.

Usability Testing

Lansdale and Ormerod (1994) break usability testing into several functions:

- Throughput: The number of objectives achieved to a specified level of accuracy within a fixed period of time.
- Execution time: The amount of time required to achieve a specified number of objectives to a specified level of accuracy.
- Accuracy: The accuracy of a specified number of objectives achieved within a fixed period of time.
- Subjective measures: Program evaluation questionnaires imbedded in interface or completed on paper, by e-mail, etc. after use of program.
- Memory Load: - Are users asked to remember too much data
- Are there excessive procedures/steps in task
- Video Tapes or logs: Videotaping user interaction or imbedded tracking systems that log the user's interactions.

The interface evaluation methodologies behind the three interactive interfaces discussed here took advantage of all of the above methodologies, but at different stages during production and during alpha and beta testing.

Modeling

Computer assisted evaluation can be used in evaluating computer assisted instruction. Several modeling tools exist:

- GOMS (Goals, Operators, Methods and Selection Rules) Card, Moran and Newell (1983) (skills necessary for GOMS and execution time estimates).
- TAG (Task-Action Grammars) Payne and Green (1986) (interface consistency).
- CCT (Cognitive Complexity Theory) Kieras and Polson (1985) (predicts ease of learning for different interface devices).
- PUMs (Programmable User Models) Young, Green and Simon (1989) (simulates users' behaviour and interaction with interface).

Although the use of modeling to evaluate interfaces shows great promise, modeling involves extremely complex tools, labour intensive procedures and large time delays that exceed the resources available for many projects. The feedback on the interface designs discussed in this thesis have been achieved with alternative methods that produce similar results.

Heuristic Evaluation

In a comparison of four evaluation methodologies, Jefferies Miller, Wharton and Uyeda (1991) found that heuristic evaluation identified the largest number of problems. Nielsen and Landauer (1993) found heuristic evaluation to be more cost effective than usability testing, and Nielsen (1992) found that heuristic evaluation had a higher probability of finding major interface problems.

The nine heuristics (rule bases) offered by Molich and Nielsen (1990) for evaluating an interface suggest that it:

1. use simple and natural language;
2. speak the users' language;
3. minimize users' memory load;
4. be consistent;
5. provide responses;
6. provide clearly marked exits;
7. provide shortcuts;
8. good error messages; and
9. prevent errors.

During the interface evaluation stages of the three interfaces discussed here, heuristic evaluation was done in-house by both an SME and a blind (objective) evaluator (see the following section on 'Alpha Testing').

Lansdale and Ormerod (1994) offer two axioms regarding the selection of evaluation methodology:

- Any approach which orients one to think about relevant issues of usability is likely to lead to useful insights.
- Evaluation methods cannot be compared or justified on theoretical grounds, but they can be assessed for their cost-effectiveness.

The two common procedures for validating all the instructional interfaces discussed here are standard alpha and beta-reviews.

Alpha-Testing

An alpha-review filters out problems unique to design, development and production. This ensures that the highest quality product is available for the beta-test. The three recommended levels of alpha-testing an interface are:

1. Spot testing done during design and production.
2. Alpha-testing done in-house prior to releasing an alpha-copy to the client.
3. Alpha-testing done by the client prior to field testing of a beta-copy.

Spot Testing

When integrating a new concept into a design, it is important to discuss novel or difficult techniques with both the graphic and programming departments. Not only do they often improve and simplify ideas, but their buy-in up-front means they are 100% on-board when it comes time to execute the ideas. Problems with a design element that are discovered later can be extremely expensive to solve.

Spot testing also estimates how much time and human resources it will take to design and deliver the interface elements. This provides valuable information for budgeting and scheduling. Sometimes it is necessary to produce a sample to get client approval and ensure that the program can be executed within the estimated time frame and cost. This also unearths any ideas that may be topical or potentially difficult to get approved. Early alpha-material should also be run on a poor dual-scan, passive matrix laptop. If it is still legible, and doesn't crash or fall apart, it will probably be glitch free once it goes into full production.

Initial samples of any layout styles, graphics, animation, photo's, and navigation mechanisms should be evaluated as soon as they become available. It is preferable to change one graphic early on, rather than 50 later. As always, any functional samples should be shipped off to the client for sign-off ASAP.

In-House Alpha-Test:

Once all elements have been programmed into a disk/disc that runs reliably, the following steps will ensure valid feedback from an alpha-evaluation.

- Never run an alpha-test from a hard-drive, server, or the programmer's platform. The seek and access times, runtime software, data paths, and files will not be the same as when run off a CD (or diskette).
- Instead, press a CD/diskette complete with install software and all auxiliary install/un-install utilities... as if it were to be shipped.
- Run the CD/diskette on a PC disconnected from any server, with none of the program elements on the host hard-drive. If any file or path is missing, or discontinuous, the program may find it and run it from somewhere else.
- The same CD/diskette should also be run on a variety of different platforms (three) that represent the minimum specified operating environment because:
  - not all operating systems are configured the same way;
  - not all Pentium processors of the same class and speed run the same way;
  - not all CD-ROM's of the same speed have the same I/O time;
  - not all graphic cards have the same memory, display and refresh rates; and
  - not all memory configurations of the same volume and speed are accessed in the same fashion.
- The CD/diskette should also be run on a laptop of the minimum specified configuration. Here is the true test of a program's durability.
- The program should be reviewed (heuristic evaluation) by an in-house quality control (QC) expert who is not familiar with the content. The mandate of QC is to evaluate the program with the proverbial 'fine tooth comb' looking for:
  - spelling errors;
  - poor grammar;
  - inconsistent use of language, definitions, icons, symbols, bullets, colour;
  - programming glitches or bugs.
- The program should be reviewed for content accuracy by an in-house SME who has not been involved in the program. The designer should be prepared to defend, explain, and rationalize all content questions.

Following these exercises, all observations and suggestions should be documented and forwarded in writing to the director and production manager who then decide what can and will be changed. The revised list is duplicated and given to the various implicated persons in the graphics, video, audio and programming departments with their 'must-fixes' highlit. This is usually done at a session where one key person from each department 'walks through' the entire program with the director. Once all changes have been made, the entire program should be checked twice by the director, in-house SME and QC.

Client Alpha-Test

Once the program is deemed 'clean', the copy used by one of the reviewers should be shipped to the client for alpha-review. Never write a new CD for the client. Errors can happen when writing a CD from the main program hard drive.

If clients are unfamiliar with the approval process, it is important to provide them with some basic training and guidelines. If not, the client's review committee may over or under-critique the program. It should also be clear what is, and is not negotiable at this point. Most projects tend to exceed the requirements of the original design, so it is rare for a program to fall short of the deliverables. However, as clients begin to understand the capabilities of the medium, their enthusiasm can translate into expectations beyond what was originally contracted. This is where the original design document drawn up at the outset of the project is helpful in comparing what was contracted, and what was delivered.

Beta-Testing

Everyone has heard horror stories about defective programs being released to market, and sometimes being recalled. With interactive training programs for the medical field, this is obviously not an option. Numerous protocols must be respected in order to conduct a beta-test in the field that is valid in its design, execution and data collection. Otherwise, any final changes made in response to the beta-review runs the risk of making modifications to the program based on invalid data collected under questionable beta-test conditions.

This is the 'last chance' to fine tune the product. It must be done carefully and correctly. The liability of releasing an imperfect product to the field is too great a risk.

Where practical, two levels of beta-tests should be conducted: first a controlled beta-test (with observation and debriefing), followed by a field beta-test (under actual distribution conditions.

Controlled Beta-Test

This is a procedure in which a small sampling of users are provided with the CD/diskette and observed while interacting with the program from installation to reporting results. Their actions are noted (informally or on video) and each subject is debriefed immediately afterwards.

The following guidelines are recommended:

- Subjects must understand that they cannot ask for help from the observer unless a major problem interferes with them completing the program.
- The debriefing questions should be well designed so as not to 'lead' the beta-subjects and bias their responses.
- Content questions should be addressed first, questions regarding the program interface, structure, and navigation tools last.
- It is important to differentiate between content and navigation issues. (e.g. did users not understand a question because of the way it was constructed, or because they were not clear on the subject matter?)

Field Beta-Test

The field beta-test is the ultimate trial of a program and all the support mechanisms associated with it such as the packaging, delivery, installation, and reporting. A lot of resources, preparation and planning go into a comprehensive field beta review, and there is never enough time to conduct a second one. To ensure a successful field beta test that produces valuable results, the following guidelines are suggested:

- A limited printing/copy of the final packaging should be run, even if only in black and white. This is especially important if install procedures, help-line numbers, and other information are included in the packaging, and/or labels.
- Beta CD/diskettes should be duplicated and packaged by the same facility that will handle final distribution. This is also an opportunity to assess turn-around time and quality control.
- The product should be delivered by the normal means (e.g. internal mail, public mail, courier, etc.). This will also provide accurate delivery delay data.
- A sample audience of 100 (minimum 60) should be selected. They should represent a good cross section sampling of the end-users.
- The sample audience should exclude any persons involved earlier in the development, alpha or beta-testing of the product.
- Outside of the performance data returned electronically or by BRC, a comprehensive standardized evaluation questionnaire should also be included, completed and returned anonymously within a specified delay.

- If possible, a sampling of the beta-audience should be debriefed in person to collect as much qualitative feedback on the interface and content as possible.
- In both the evaluation questionnaire and debriefing, it is important to distinguish clearly between questions regarding the program interface, structure, navigation tools, etc., and questions regarding content.

Beta-testing in the real world.

Because of unforeseen delays, production schedules have the habit of shrinking. Beta-testing is one of the last steps in the entire operation, prior to release. As a result, beta-testing is often compromised, and sometimes moved into the roll-out release stage as a 'limited release' within the client's immediate environment.

This compromise amounts to conducting no beta-test at all. Without proper evaluation methodology, the data returned may be of questionable validity, or even worse, misleading. And if the beta-test has been canceled as a result of schedule compromises, the period required to consolidate useful beta-test data and make changes to the program will probably be lost as well.

As a result, the beta-test protocol, client involvement, and execution period should be defined and agreed upon during the contracting stage because:

- It alerts the client to the importance of the beta-test process in guaranteeing the highest possible quality and effectiveness of their product.
- It gets client buy-in from many levels early in the process. This gives the people and departments implicated ample time to arrange the necessary resources to facilitate the beta-test procedure.
- When clients know they will be involved in the beta-testing procedure, it generates internal interest and helps drive distribution by end-point user interest.
- It involves the client in the design, execution and correlation of results. Therefore it is unlikely that the objectivity, design and methodology of the evaluation procedure will ever be questioned.

Evaluation vs. Approval and Testing

If there is a fundamental difference between the alpha evaluation of a product during its production and approval and testing during the beta-release stage, that difference is objectivity. Having already guided a product through the approval and testing stages, a designer may ask, "What tools are available during the beta-evaluation that could not have been used during the various stages of production?"

One conclusion is that evaluators must select and employ the tools that they feel will best measure the interface, but with a methodology that both the budget and schedule can accommodate.

Program Performance Tracking

> *"...survival has relied upon our ability to learn and adapt. It is my premise, that interfaces should as well."*
>
> Lansdale and Ormerod (1994)

One powerful and promising set of tools that can be imbedded into the interactive interface are tracking functions. These sub-routines assemble user-interaction data for the sole purpose of the evaluation and improvement of the program design itself, measuring the program's performance as opposed to the users' performance. This involves recording the users' progress through the program, generating data that are then transmitted to a central reporting facility, along with data on the users' performance. Such data can provide objective information on the performance of the program's interface as measured by users' interaction with the navigation and learning tools.

While showing great promise, this mechanism presents several problems:

- There are substantial, additional programming costs to embedding program performance tracking into an interface.
- Program performance tracking may only be used in beta-test because the data output is too large to be encoded for all remote reporting mechanisms except background e-mail.
- Analyzing the extensive data generated by program performance tracking is an extremely labour intensive procedure.
- Correlating program performance tracking data with users performance data and quantitative evaluation data is a massive task. Because of the time factors, results can only be applied to interfaces in later programs.

In some cases, such as with the X-Plore series, specific metrics were tracked, such as quantitative evaluation and user performance scores reported by specific geographic regions, or the perceived prescribing habits of doctors.

However, to date, the author has not had the opportunity to design a protocol that would provide meaningful results from the extensive data that can be generated by detailed tracking (a record of every single action of each user within the interface from sign-in to exit).

Lansdale and Ormerod (1994) recognize that although tracking systems provide insight into the use of interfaces, there are also simpler more effective methods of collecting the same data. Because of limited time and resources, most designers and clients must use the methods that are the least expensive. However it is only a matter of time before complex and intelligent algorithms that can convert system logs into useful information become cost effective. For example, if the tracking revealed that a person had a higher frequency of use of one particular help menu function, this could be a reflection of:

- the effectiveness of the design of the help tools;
- that user's learning style;
- that user's familiarity with navigation tools;
- that user's familiarity with the content; or
- numerous personal, historical, or cultural biases in the interpretation of the icon representing that tool.

In order for detailed tracking data to be useful, it is necessary to select, then standardize a sample user population based on their learning styles, familiarity with computer interfaces, familiarity with a standardized neutral content. Only then does data from tracking become useful in making decisions regarding modification of program design.

Step 10: Results

The Evolution of Evaluation

The interactive interfaces selected for this discussion demonstrate three different approaches to evaluation.

The X-Plore series was designed primarily as an entry level evaluation tool to measure prerequisite knowledge of the sales force of a particular pathology, and their response to CAI. The Backgrounder series included a comprehensive instructional component, followed by a practical application tool (labs and case studies), and finally a 'Final Exam' to measure exit performance. The Pacing Primer was a departure from the previous two interfaces in that performance evaluation was not a critical endpoint. The final 15 question exam did cover fundamental content, but was basically a value added component because it offered CE credits. In this sense, the purpose of the exit evaluation was to encourage users to explore the instructional content in order to achieve a passing grade (80%), rather than to demonstrate proficiency.

The X-Plore Series Results

The X-Plore interface was used in two series: a three module pilot series, X-Plore CHD, and the six module X-Plore Hypertension series. Results were reported via a BRC card. (See the previous section on Tracking and Reporting).

Results from the pilot X-Plore CHD series are not available. Results from Modules 3 and 4 of the X-Plore Hypertension series are also not available. Results from Modules 5 and 6 of the X-Plore Hypertension were combined. Results from the X-Plore Hypertension series are as follows:

Module	**1**	**2**	**3**	**4**	**5 & 6**
Respondents	723	656			413
Mean Score	74.4%	87.6%	N/A	N/A	89.3%
Standard Deviation	13.8	10.3			11.2
Average Time (min.)	71-80	56-60			36-45
Knowledge	4.0	4.1			4.2
Maintain interest	3.7	3.9			4.0
Time well spent	3.7	3.9			3.9
Receive more	3.6	3.8			3.9
Overall appreciation	3.8	3.9			4.0

Table 17: Results from X-Plore Hypertension series

Key

- Score represents the mean overall performance score based on the number of questions correctly answered on the first pass before remedial activity.
- Standard Deviation of the population from the mean score.
- Avg. Time represents the average time-on the system per module.
- Knowledge: "Did the module help you assess your general knowledge of hypertension?" (all range 1-5, where 5 is best).
- Maintain Interest: "Did the program maintain your interest throughout?"
- Time well spent: "Was your time on this module well spent?"
- Receive more: "Would you like to receive future training material in this format?
- Overall appreciation is an average of the four appreciation scores.

Discussion

Users' territory codes were cross-referenced with the employee database to extract demographics such as performance (mean scores) by region, territory, sales performance, and other comparative analysis that was of value to the client. As expected, average time-on decreased through the series.

From the designer's perspective, the most valuable data was the qualitative write-in responses. Although difficult to classify it proved useful to solve minor interface problems. (E.g. in an earlier series, it had been identified that there could be more than one correct response. This repetitive statement took up screen space so was not repeated in later modules, except in the navigation instructions. However, some users reported scoring incorrectly because they were not alerted that there could be more than one correct response).

It was generally observed that the quality of the content of the second module in the series had improved since the first module. This was an accurate observation since more time had been available for the development of the second module. Although difficult to measure, these observations may have also been a function of content, which evolved module by module from fundamental physiology and pathology in the first module, to more practical issues such as diagnosis and treatment strategy in the last. To date, the relationship between qualitative assessment of content and content evolution has not been studied in these products.

Qualitative Data

The following is a sampling of the type of qualitative data that was received from the write-in response area:

- *"Excellent program! Provides valuable feedback in a user-friendly design. Good pathways for learner remediation. Very innovative and well designed."*
- *"The last 3 sections of testing were interesting and actually fun to do."*
- *"Excellent format for reviewing background information. Easy to follow, fun and a good memory refresher."*
- *"This is much better than reading and taking a test later. It seemed a lot more interactive and thought-provoking."*
- *"Good format for self-paced learning."*
- *"I enjoyed the training program setup - very hands on!"*
- *"The last two sections were fun to complete - the others were, to say the least, thought provoking."*
- *"I enjoy this format of training."*
- *"More practical than Module 1. "Who Am I" section very helpful."*
- *"I like the format of the study materials, and the test via computer. Congratulations on this project."*
- *"Good format. These programs are great. Instant feedback is best for real learning."*
- *"I really liked the "What Drug Am I?" part, it was really fun and a nice way to end the tests."*
- *"Well done (very user friendly). We need to do something like this for HMGs."*
- *"Very interactive and fun to use, maybe give final grade with completion code."*
- *"It was a lot of fun working with this module. Very interactive."*
- *"This is by far one of the most beneficial modules that I have ever seen or studied. Excellent understandability and great organization. Great job!"*
- *"Again, excellent. However I needed a definition of the following: COPD and hyperucimia. A great training tool."*
- *"I especially liked the patient profiles."*
- *"The feature that allows to exit and re-enter at same starting point was great!"*

The Backgrounder Series Results

Three series employed the Backgrounder interface: Hypertension (5 discs); Asthma (4 discs); and Male Androgenetic Alopecia - (5 discs). A 'Final Exam' concluded each module, and seven program evaluation questions concluded the series. Results were reported by background e-mail. (A sample of the reporting screen is in: Tracking and Reporting). At the time of this submission, results for the Hypertension Backgrounder series were not available.

Asthma

Module	1	2	3	4
Respondents (all)	1374			
Score (mean/module)	85.5	93.3	84.6	89.7%
Standard Deviation	13.2	7.9	12.6	9.2
Objectives *	3.5			
Interest *	3.3			
Graphics *	4.1			
Narration *	3.5			
Navigation *	3.3			
Labs *	3.2			
Evaluation *	3.1			

Table 18: Results from Asthma Backgrounder series

Male Androgenetic Alopecia

Module	1	2	3	4	5
Respondents (all)	343**				
Score (mean/module)	96.7%	97.5%	96.04%	97.69%	95.70%
Standard Deviation	5.9	5.3	6.7	5.4	5.9
Objectives *	3.8				
Interest *	3.4				
Graphics *	4.0				
Narration *	3.5				
Navigation *	3.4				
Labs *	3.3				
Evaluation*	3.3				

* Mean scores, all modules. ** Number of respondents at time of submission

Table 19: Results from Alopecia Backgrounder series

Key

- Score represents the score on the final exam at the end of each module.
- Standard Deviation of the population from the mean score.
- Objectives: "Where you able to achieve the learning objectives stated in the program?" (all range 1-5, where 5 is best)
- Interest: "Did the program maintain your interest level throughout?"
- Graphics: "How would you rate the quality of the graphics?"
- Narration: "How would you rate the quality of the narration?"
- Navigation: "How would you rate the ease of the navigation?"
- Labs: "How would you rate the quality of the lab exercises?"
- Evaluation: "How would you rate the quality of the evaluation questions?"

Discussion

The rating of elements related to the instructional components of the program (objectives, interest, labs) improved from the Asthma series to the Alopecia series. This may reflect the instructional vehicles that were used for the respective modules. Technical issues such as the quality of the graphics and narration rated consistently high. Of interest is that the opinion of the quality of the evaluation questions was rated differently for the two series, although the respondents, evaluation screen and questions were identical. The same applied for the navigation, which, although identical for the two series, received different ratings as well. The implication of the data from the Backgrounder series will be studied once results from the third program in the series (Hypertension) have been received

The Pacing Primer Series Results

At the time of this submission, collection of the Pacing Primer series results had just begun, therefore no data was available. Instead, the results from the demo trial of the interface are being included as a demonstration of the type of feedback that can be collected to guide in the development of an interface.

A six minute demo version of the interface was shown to 46 critical care nurses who completed an evaluation form and were debriefed following their trial of the demo program. (For more details on the demo procedure, see the preceding section on 'Testing a Demo').

Discussion

The program, without exception, was highly rated. Both concept and execution were embraced with enthusiasm, interest and a hunger for more.

Almost all respondents felt strongly that such a program would work quite well as a learning tool for them, with 95% preferring to learn about pacing using computer-based training. In fact, when asked whether they thought their institution should adopt a computer-based learning tool for pacing such as the one they tested, agreement was almost unanimous (98%) that it should.

Program features were highly praised, in particular ease of use, clarity of instructions, 'fun' in the learning process, relevant medical content, simulation of real-life situations, and the opportunity to earn contact hours for training.

Criticisms were offered, mostly pertaining to specific details unrelated to the merit of the program in general (e.g. no way to go back to previous page to review, a limitation unique to the demo program).

Most respondents agreed fairly strongly that computer-based training saves time. A number of educators indicated that it would help to ease their burden by providing a controlled learning tool which trains and monitors. Some felt that whereas time-savings may not be a factor, the self-paced feature added flexibility.

Most nurses (88%) reported having access to a computer at work although less than half the computers were located in an area conducive to learning. A little over one third of those with access to computers knew whether they had CD-ROM capability. Most respondents (81%) had a computer at home and about 70% of home-users had CD-ROM capability.

Notably, two thirds of nurses said they would be most likely to use computer-based training at home, while the rest said they would prefer their workplace. Almost all respondents (93%) indicated they would like to have personal follow-up by a company representative.

Summary

Results from interactive computer programs address the needs of three groups: the client, the designer, and the users. Clients demand measurable data demonstrating that the desired changes in users' behavior, ability or knowledge have taken place. The data reported in the 'Results' section of this document are indicative of what clients expect. And the needs of the client must be served first and foremost.

Concurrently, interface designers have their own agenda. Designers want to engineer results that support existing instructional strategies imbedded in the interface, but that promote future potential strategies as well. If the time, resources, and disc space were available (and reporting mechanisms capable), each interface would be loaded with tracking mechanisms collecting reams of data on how users interacted with the numerous tools, and their supporting cognitive, behavioral and instructional theories.[12]

The section on 'Tracking and Evaluation' addressed the importance of gathering data for use in the interface design of subsequent programs. Due to the delay between the collection of data and its application, benefits can usually only be applied to a specific user group, over a period of time. However, as it is recorded, the same data should also be used to adjust the program content and

---

[12] To illustrate this point, imagine if the same question could be asked using four different evaluation mechanisms across four versions of the same interface, or if four different presentation techniques could be used, but the same question asked. The results would establish powerful precedents for future interfaces.

learning strategies to each user's individual ability and preferences. And this should happen in real-time as a user progresses through the program. The feedback cycle should be immediate, and individual.

This implicates the third group, the users. Some of the users' needs are similar to those of the client, (although perhaps they are more immediate in that all users want feedback on their performance in real-time, or at least, by the end of the program). But the real-time measurement of performance has other implications that may not be so obvious to the users. Real-time results can be used to determine to what depth (detail or level of difficulty) and breadth of content (scope of subject and associated material) users should be exposed.

For example, in the Sherlock 1 System (Goodman 1993) the program provides five different levels of instruction and/or explanation which vary in detail, depth and duration, depending on a user's ability as assessed by the system. This is similar to the concept of a user-centered interface in which the interface exercises control, deciding to what level of content depth users should be exposed.

On the other hand, the Interactive Pacing Primer interface provides a different form and locus of control. Users determine the breadth and scope of the information they wish to interact with. In future interface designs, the author hopes to explore a protocol that enables the <u>interface</u> to make these decisions, enabling users to focus more on content synthesis than on content selection.

Lynch and Horton (1997) see such mechanisms as 'agents' that can track users' activities as they move through the interface in order to predict what their instructional or navigational needs will be. The system then adjusts the interface to provide each user with the tools most appropriate for their learning style or preference.

Lansdale and Ormerod (1994) suggest that the ability to create self-adjusting, self-guided learning tools (that bring CAI one step closer to the true concept of interactivity) will depend on the extent to which interfaces can be designed to analyze each user's learning style, then respond to their needs with the appropriate learning tools and strategies.

Ultimately, it becomes a multi-tiered issue of who should have control: the interface, the users, or both? Who should decide: the interface or the users? And what are the conditions of those decisions?

This is an area that the author hopes to investigate in future studies.

References

Baeker, R., Small, I. (1990). Animation at the interface. In B. Laurel (Ed.), The Art of Human Computer Interface Design, (251-267). Reading, MA: Addison-Wesley.

Brown, J.R. & Cunningham, S. (1989). Programming the User Interface, Principles and Examples, New York: John Wiley & Sons.

Card, S.K., Moran, J.P., & Newell, A. (1983). The Psychology of Human Computer Interaction, Hillside, NJ: Lawrence Erlbaum Associates. (41-97).

Carliner, S. (1990) Cited in J. Fitz (Author). Conquering the dreaded electronic textbook. Training: the Magazine of Human Resources Development, 27 (5) 15.

Carrol, J.M., Olson, J.R., Kellog, W.A. (1988). Interface, Metaphors, and User interface Design. Handbook of Human Computer Interaction, M. Helander, (Ed.), Amsterdam: Elsevier Science Publishers. (67-85).

Chignell, M.H. & Hancock, P.A. (1988). Intelligent Interface Design. In M. Helander (Ed.), Handbook of Human-Computer Interaction, Amsterdam: Elsevier Science Publishing. (969-995).

Cole, I., Lansdale, M.W., & Christie, B. (1985). Dialogue design guidelines. In B. Christie (Ed.), Human Factors of Information Technology in the Office, Chichester: John Wiley & Sons. (212-241).

Doughty, R.K., Kelso, J. (1984). An evaluation of menu width and depth on user performance, Unpublished project paper done with Prof. James Foley, George Washington University, Washington, DC: Cited in B. Shneiderman Designing the User Interface: Strategies for the Effective Human-Computer Interaction, Reading, MA: Addison-Wesley Publishing. (99).

Desmarais, N. (1994). Multimedia on the PC: A Guide for Information Professionals, New York, NY: McGraw-Hill.

Dray, S.M., Ogden, W.G., Vestewig, R.E., (1981). Measuring performance with a menu-selection human-computer interface. Proceedings of Human Factor Society, 25th Annual Meeting, (746-748).

Eason, K.D. (1988). Information Technology and Organizational Change, London: Taylor and Francis. (129-156).

Floyd, S. (1982). Thinking Interactively. In S. & B. Floyd (Eds.), Handbook of Interactive Video, New York: Knowledge Industry Publications. (3-8).

Gagné, R.M. (1985). The Conditions of Learning and Theory of Instruction, New York: Holt, Rinehart and Wilson.

Goldberg, R (1996). The Multimedia Producer's Bible, Foster City: IDG Books Worldwide.

Goodman, B.A., (1993). Multimedia Explanations for Intelligent Training Systems. In M.T. Maybury (Ed.), Intelligent Multimedia Interfaces, Cambridge, MA: MIT Press. (148-173).

Green, A.J. K. & Gilhooly, K.J. (1990). Individual differences and effective learning procedures: the case of statistical computing. International Journal of Man-Machine Studies, 33, (97-119).

Hampton, J.R. et al. citing Tudor, J.H. (1975). Relative contributions of history-taking, physical examinations, and laboratory examination to diagnosis and management of medical outpatients. British Medical Journal, 2 (486-489).

Heckel, P. (1984). The Elements of Friendly Software Design, New York: Warner Books.

Hon, D. (1982). Future Directions: The curtain rises on interactive video. In S. & B. Floyd (Eds.), Handbook of Interactive Video, New York: Knowledge Industry Publications. (125-130).

Info-Line. (1987). Get Results with the Case Method. American Society for Training & Development. Issue 703.

Jefferies, R., Miller, J.R., Wharton, C., & Uyeda, K.M. (1991) User interface evaluation in the real world: A comparison of four techniques. In Proceedings of the ACM Conference on Human Factors in Computing Systems, New Orleans: ACM Press. (119-124).

Johanassen, D.H. (1985). Interactive Lesson Design: A Taxonomy. Educational Technology, 26, (6), (7-16).

Kamouri, A.L., Kamouri, J., & Smith, K.H. (1986). Training by exploration: facilitating the transfer of procedural knowledge through analogical reasoning. International Journal of Man-Machine Studies, 24, (171-190).

Kieras, D.E., & Polson, P.G. (1985). An approach to the formal analysis of user complexity. International Journal of Man Machine Studies, 22, (365-394).

Lansdale, M.W. & Ormerod, T.C. (1994). Understanding Interfaces - A Handbook of Human-Computer Dialogue, London: Academic Press, Harcourt Braces & Co.

Larkin, J.H. (1989). Display-based problem solving. In D. Klahr & K. Kotovsky (Eds.) Complex Information Processing, Hillsdale, NJ: Lawrence Erlbaum Associates. (319-341).

Laurel, B.K. (1986). Interface as Memesis. In D.A. Norman & S.W. Draper (Eds.), User-Centered System Design, Hillsdale, NJ: Lawrence Erlbaum Associates. (67-86).

Lewis, C. Polson, P.G., Wharton, C., Rieman, J. (1990). Testing a walkthrough methodology for theory-based design of walk-up-an-use interfaces. In Proceedings of the ACM Conference on Human Factors in Computing Systems, Seattle, WA: ACM Press. (235-242). (abstract)

Lynch, P. Horton, S: Yale C/AIM Web Style Guide, (1997). Yale University .([on-line] available: http://info.med.yale.edu/caim/manual)

Lynch, P.J, (1994) The Evolving Interface of Multimedia. Syllabus Magazine, 8 (3), (48-50).

MacGregor, J.N., Lee, E.S. (1987) Performance and preference in videotext menu retrieval: A review of empirical literature. Behaviour and Information Technology 6 (1), (43-68). Cited in M.W. Lansdale, & T.C. Ormerod Understanding Interfaces - A Handbook of Human-Computer Dialogue, London: Academic Press, Harcourt Braces & Co. (48).

Maul, Gary P. & Spotts, David ,S. (1993). Developing computer-based instructional courses Industrial Management, 35 (6), (1-9).

Meggs, P. B. (1989). Type and image: The language of graphic design. New York: Van Nostrand Reinhold.

Miller, D.P. (1981). The breadth/depth tradeoff in hierarchical computer menus. In. Proceedings of Human Factors Society, 25th Annual Meeting, (296-300).

Miller, G.A. (1956). The magical number seven plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63 (81-97).

Molich, R. & Nielsen, J. (1990). Improving a human-computer dialogue. Communications of the ACM, 33 (3) (338-349).

Myers, B.A., & Rosson, M.B. (1992). Survey of User Interface Programming. Proceedings of ACM Conference on Human Factors in Computing Systems, Monterey, CA: ACM Press. (195-201).

Mynatt, B.T., Leventhal, L.M., Instone, K., Farhat, J. & Rohlman, D. (1992). Hypertext or Books: Which is better for answering questions? Proceedings of ACM Conference on Human Factors in Computing Systems, Monterey, CA: ACM Press. (19-25).

Nielsen, J. (1992). Finding Usability Problems Through Heuristic Evaluation. Proceedings of the ACM Conference on Human Factors in Computing Systems, ACM Press. (373-380).

Nielsen, J., & Landauer, T.K. (1993). A mathematical model of the findings of usability problems. Proceedings of the ACM Conference on Human Factors in Computing Systems, Amsterdam: ACM Press. (206-213).

Norman, D.A. (1990). The Design of Everyday Things. New York: Basic Books.

Norman, D.A. (1986). Cognitive Engineering. In User-Centered System Design: New Perspectives on Human Computer Interaction, Norman, D.A. and Draper (Eds.), Hillsdale, NJ: Lawrence Erlbaum Associates. (31-62).

Oz, E. & White, L. D. (1993). Multimedia for Better Training. Journal of Systems Management, 44 (5), 34.

Payne, S.G., & Green, T.R.G. (1986). Task-Action Grammars: A model of mental representation of task languages. Human-Computer Interaction, 2 (93-133). Cited in M.W. Lansdale, & T.C. Ormerod Understanding Interfaces - A Handbook of Human-Computer Dialogue, London: Academic Press, Harcourt Braces & Co. (114).

Perez, L. D.& Willis, P. H. (1989). CBT products improve training at reduced cost. Computers in Healthcare 10 (7), 28.

Roth, S.F., & Hefley, W.E. (1993). Intelligent Multimedia Presentation Systems: Research & Principles. In Maybury M.T. (Ed.) Intelligent Multimedia Interfaces, Cambridge, MA: MIT Press, (13-59).

Savoie, M. (1994) Testing the Effectiveness of an Automated Tool to Evaluate the Ease of Use and the Ease of Learning of a Computer Interface. Unpublished manuscript, Montreal: Concordia University, Faculty of Commerce & Administration. 8.

Shneiderman, B. (1987). Designing the User Interface: Strategies for the Effective Human-Computer Interaction, Reading, MA: Addison-Wesley.

Shneiderman, B. (1997). Direct Manipulation for Comprehensible, Predictable and Controllable Interfaces. In Proceedings of the 1997 international conference on Intelligent User Interfaces, ACM Press. (33-39).

Smith, J. (1983). How to Design Interactive Training Programs, Training, 20,(12) (30-45).

Smith, S., Bergeron, R.D. & Grinstein, G.G. (1990). Stereophonic and surface sound generation for exploratory data analysis. In Proceedings of the ACM Conference on Human Factor in Computing Systems Seattle, WA: ACM Press. (125-132).

Webster's Encyclopedic Dictionary, Canadian Edition, (1988). s.v. Interface. Canadian Edition

Whiteside, J., Bennett, J., & Holtzblatt, K. (1998). Usability engineering: Our experience and evolution. In M. Helander (Ed.), Handbook of Human-Computer Interaction, Amsterdam: Elsevier Science Publishing. (791-816).

Young, R.M. (1983). Surrogates and mappings: Two kinds of conceptual models for interactive devices. In Getner, D., Stevens, A.L. (Eds.), Mental Models. Hillsdale NJ: Lawrence Erlbaum. (35-52). Cited by M.W. Lansdale & T.C. Ormerod. (1994). In Understanding Interfaces: A Handbook of Human Computer Dialogue. London: Academic Press, Harcourt Braces & Co. (16).

Appendices

Appendix 1: An Interface Design Checklist

Constantly check the interface design against this list of design fundamental do's and don'ts.

1. Minimize text, maximize graphics and visual analogies
2. Present information (across all media) in small, easily digestible bites.
3. Respect visual design principles of colour, font, size, spacing, margins, density, layout, etc.
4. Keep all screens simple, logical and attractive.
5. An interface tends to grow in complexity exponentially.
6. Make even mundane activities interesting, novel, fun.
7. Be consistent. Establish and adhere to standards.
8. Keep levels/layers to a minimum. Don't loose the users in structure.
9. Navigation should be bi-directional or omni-directional.
10. Let users know what they have done, what they still have to do (where they are, where they have been, where they can still go).
11. There should be lots of feedback, and it should be meaningful.
12. Avoid extensive branching. Your program flowchart should be lean.
13. Select an interface analogy that is universal to both the subject and the audience.
14. Use functional, universal icons that are apparent without labeling or explanation.
15. Never sacrifice functionality and efficiency for artistic or entertaining objectives.
16. Understand and keep the target user's cognitive preconceptions in sight.
17. Each audience has its own language and culture. Study and mimic it.
18. Before committing to them, research and test all tools on a laptop of the lowest common denominator of your target hardware install base.

Despite all of the techniques, tools and creativity that can be tapped into to create an interface, as one author put it succinctly "...a good design will still be a good design, even with the limitations of existing resources. (Lynch, 1994)

Appendix 2: Disc 'Real Estate'

When CAI programs were being delivered on 1.4 MB floppy diskettes, it forced the designer to be very judicious with instructional strategies, and very creative with the limited material that would fit on one or more diskettes. One would presuppose that with 660 MB on a CD, this would no longer be of concern.

Despite the fact that better graphic cards and compression codecs can display higher resolution, million colour graphics, and run larger video windows with higher resolution at faster frame rates, several factors have led to 'real estate' problems on interactive CD's:

- Program layers, branching and depth create more options.
- User expect more audio, animation and video in instructional tools.
- Users are accustomed to full screen, full motion video at higher colour depth.
- Screen layout is becoming increasingly more artistic and complex.

The underlying problem is allowing the medium to take precedent over the content and objectives. This problem is well illustrated by the prevalence of ShockWave® and Java® applets on the www. The introduction of new 100:1 compression codecs for web-based instruction file-transfer, and the standardization of DVD protocols, promise to alleviate this problem. However, although a complex interface may be attractive, more information often implies more 'noise', less content, and possibly a lower retention rate.

As demonstrated by Goodman, (1993) among interactive tools, video is ranked lowest in expected overall effectiveness when compared with five other courseware media.

Projecting 'real estate' requirements

When selecting and designing instructional tools, keep a rough 'balance sheet' with average file volume sizes (on a spreadsheet for update and 'what-if' scenarios). This will avoid designing in more than what can fit on one CD. It is a best practice to create one sample for each tool (if possible, one that has been approved by the client), then measure it and use it as your baseline. For planning and projection purposes, use a 1-Σ deviation until the actual production samples comes out from the various departments. Then keep a running tabulation comparing actual file volumes x actual number of files, compared to projected file volume x number of files.

File Type	Criteria	Size	Units	Sub
Text	W/P or authoring file	10 k	full screen	
Graphic	pict/tiff resource file	100 k	full screen	
Photo	JPEG/GIF file	250 k	full screen	
Text, graphic, and/or photo	bitmap	350 k	full screen	
Audio	8 bit, 22khz	1.3 M	per minute	
Animation/video	1/8 screen (160 x 120)	2.3 M	per minute	
Animation/video	1/6 screen (240 x180)	3.7 M	per minute	
Animation/video	1/4 screen (320 x 240)	5.5 M	per minute	
Animation/video	full screen (640 x 480)	22 M	per minute	
QuickTime player		1.7 M	one per disc	
Authorware runtime		700 k	one per disc	
Install/un-install software	install only / both	700 k /1.M	one per disc	
Program	varying	1% to 10%	total pgm. vol.	
				Total

(*Approximate values for estimation purposes. Actual values will differ)
(**Reference values: 8 bit, 256 colours, 4X CD data rate 250kbps)

Table 20: Total file volume estimation

Multi CD-ROM Series

660 MB on a CD seems like an infinite amount of space, especially after designing interactive training for delivery on floppy diskettes. However, with the demand for audio, video and animation in instructional delivery, a CD soon becomes very small-- sometimes too small.

Should initial estimates of the disc real estate requirements of a program exceed the 660 MB barrier, multi-CD delivery is always an option. Multi-CD programs are commonplace. Unless the user-base has a CD carousel (which still have a substantial delay), multi CD programs require that the information be broken up into modules-per-disc. While this has the advantage of potentially limitless file volume space, there are some limitations:

Some multi-disc limitations affect only the amount of available space per disk (unless they are installed on the hard drive):

- Tools common to all disks such as hypertext databases, lexicon's, resources, etc. must be placed on each disc.
- Text, audio, video and graphic files that are shared by various modules must be placed on each disc.

Some multi-disc limitations affect the program design itself.

- Users cannot navigate to or from <u>anywhere</u> in the global program, only within the disc or module.
- If the user-base does not have a CD carousel (local or server), the pause to change CD's adds delays to navigation, interaction and learning.
- In some cases, the size of common resources will be too large to justify installing them on each disk, and will only be accessible on one disc.
- Users must run the series in sequence, otherwise the install, tracking, and multi-user routines become more complicated to program.

Digital Versatile Disc

The introduction of Digital Versatile Disc (DVD) will solve the multi-disc problem. With a storage capacity of 4.7 GB (single sided, single layer) up to 17 GB (double sided, double layer) DVD offers the rough equivalence of 7.7 to 26 standard CD-ROM's. And while the industry struggles to agree on a standard -- and the authoring and install base upgrades to DVD players -- compression codecs and data transfer rates will continue to improve. This will allow for even greater storage of higher quality 3D graphics and animation, audio and video with faster access times to larger files.

One problem can be expected as interactive training migrates to the DVD platform. While unlimited disk space would remove restrictions on general quality issues such as colour depth, still and moving image resolution and size, there will also be a tendency for interfaces to focus on style and effect, with less attention given to proven, effective presentation methods supported by sound instructional strategies. This liability is repeatedly recognized by Lansdale and Ormerod (1994) and Lynch and Horton (1997)

There is much which is valuable to be learned from the challenge of developing interactive instructional material on one or more disks with a limit of 1.4 MB

Glossary

AV: audio-visual

AVI: Microsoft video display protocol

'Binders': traditional instructional systems where all elements (instructor guides, trainee workbooks, overhead acetates, videotapes, etc.) are assembled in one binder.

BRC: business reply card

CAI: computer assisted instruction

CBT: computer-based training

CD: compact disc

CE: continuing education

CHD: coronary heart disease

CRT: cathode ray tube (computer display tube)

disc: CD-ROM disc

disc real estate: the amount of space available on fixed or removable media

diskette: 3.5" floppy diskette

DVD: digital video disc

ECG: electrocardiogram

firewall: a security barrier that separates a companies information systems from unauthorized external access.

GUI: graphical user interface

HTML: hypertext mark-up language

ICW: interactive courseware

Intranet: internal Internet-type LAN or WAN positioned behind the firewall of an organization

LAN: local area network

LCD: liquid crystal display

OCR: optical character reading

QC: quality control

QT: QuickTime video display protocol

SME: subject matter expert

UCSD: user-centered system design

WAN: wide area network

WWW: worldwide web

I/O: input/output

Copyright