

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF TORONTO

CSC318S

**THE DESIGN OF
INTERACTIVE COMPUTATIONAL MEDIA**

Lecture 16 — 16 March 1998

THE EXTENDED INTERFACE — USER SUPPORT WITH
TRAINING, DOCUMENTATION, ERROR HANDLING,
AND HELP SYSTEMS

16.1 Introduction.....	2
16.2 Why do users have trouble?	3
16.3 How do users learn.....	5
16.4 Teaching and training	5
16.5 Documentation	6
16.6 Minimalist design and active learning	7
16.7 System dynamics and response time	7
16.8 Designing for human error	8
16.9 System-provided help for users	9
16.10 Human help systems	11
16.11 Individual differences.....	11

Ronald Baecker
Professor of Computer Science,
Electrical and Computer Engineering, and Management
University of Toronto

Copyright © 1991-1995, 1998, Ronald Baecker.
All rights reserved.

16.1 Introduction

Why some systems work for their users and others do not

Appropriate functionality

Designed to meet real user needs

Supportive, receptive organizational context

No major social or political resistance, roadblocks
(To be discussed in CSC428)

Appropriate Designers' Model effectively translated
into System Image,

facilitating development of an effective
Users' Conceptual Model,
thus making system easier to learn

Appropriate, congenial interaction techniques –
the right transducers,

attractive, well-designed displays,
the right dialogue style, well programmed –
thus making system easier to use

But this is not enough:

System must be learned

Users must be assisted and supported when confused,
lost, or in trouble

Thus we must enhance the overall system useability –
provide what is now being called *user support* in an
extended user interface

16.2 Why do users have trouble?

Carroll (BB, pp.639-648) argues that users learning the game of Adventure and those learning a typical serious application (like text editing) have similar problems:

Disorientation

The user/player doesn't know what to do in the system environment, hence gets lost

Illusiveness

The system seems not to do what he/she wants

Emptiness

The system is mute, unhelpful; screen provides no hints as to what to do or what went wrong

Mystery message

Feedback, when it comes, is useless or misleading

Slipperiness

Behaviour appears unpredictable, inconsistent; doing the "same thing" in different situations has unexpectedly different consequences

Side effects

System does hidden actions with later adverse repercussions and consequences

Paradox

System seems wrong, suggests actions that are clearly inappropriate

Laissez-faire

System doesn't provide guidance or support towards the achievement of overall goals

Mack, Lewis, and Carroll (BB, pp. 269-277), studied the learning of word processing, and concluded:

Learning is difficult

Learners experience frustration and blame themselves.

Learning takes longer than expected; learners have trouble applying what they know.

Learners lack basic knowledge

Learners are naive about how computers work (e.g., do not understand computer jargon).

Learners do not know what is relevant to understanding and solving problems.

Learners make ad hoc interpretations

Learners try to construct interpretations for what they do or for what happens to them.

Learners' interpretations can prevent them from seeing that they have a problem.

Learners generalize from what they know

Learners assume that some aspects of text editors will work like typewriting (especially functions that simply move the typing point on a typewriter).

Learners have trouble following directions

Learners do not always read or follow directions.

Learners do not always understand or correctly follow directions even when they do try.

Problems interact

Learners have trouble understanding that one problem can create another.

Interface features may not be obvious

Learners can be confused by prerequisites and side effects of procedures.

Learners can be confused by feedback messages and the outcome of procedures.

Help facilities do not always help

Learners do not always know what to ask for.

Help information is not always focused on the learner's specific problem.

16.3 How do users learn

Carroll and Mack (BGBG, pp. 698-717) describe how users learn computer systems, yielding further insight into the learning process

Learning by doing

- Desire to try things out

- Tendency to jump the gun

- Difficult in following written sequences of instructions

Learning by thinking

- Attempt to construct reasonable interpretations, proper *mental models* (sense-making)

- Purposeful problem solving activity

Learning by knowing

- Ability to make use of prior knowledge

 - From *metaphors*

 - From past work experience

16.4 Teaching and training

Problem – How to get started

Solution – Initial teaching and training

But students are at different levels with different learning speeds

- Key problem: How to provide ongoing learning

 - opportunities after some mastery has been attained

- How to facilitate ongoing learning by doing, thinking, knowing

16.5 Documentation

Problem – Teachers, helpers not always available, or consulted with reluctance – Exploratory learning hits bottleneck

Documentation – Written materials, can be consulted at will
Tutorial guides for beginners
Principles of operation manuals
Reference manuals for advanced users
Command summaries

The need for and importance of lots of examples!!!

The art of document design
Need to consider (Wright, BB, pp. 613-620)
How readers search through documents
How readers understand
How readers apply understanding
Document organizers
Document form (appearance)

Need for testing, debugging, *quality assurance* of documents

Need for integration of documentation with training materials, error messages, help systems

But, despite these ideas, documentation is little used and typically scorned, often for good reason

16.6 Minimalist design and active learning

Problem – Hard to structure training and documentation so it handles needs of all kinds of users

Try to anticipate all situations, all problems, but result is more and more complexity, and still can't do it

Carroll (BB, pp. 621-626) argues that we should go to the opposite extreme, encourage active learning by...
Giving users less (*minimalist design*)

Minimal manual – terse, modular, encourages exploration

Training wheels processor – Disable complex or dangerous functions, let the user play

The need for *undo*

16.7 System dynamics and response time

Need for rapid response rather than sluggish behaviour

Importance of response consistency, dangers of variability

Importance of feedback, % done progress indicators

16.8 Designing for human error

Minimizing error

Avoiding misunderstandings via training, documentation

Giving feedback re modes, legal commands

Appropriate representation

E.g., allow selection from small numbers of alternatives where possible, instead of specification

Automatic correction where possible (e.g., spelling)

The importance of *undo*

Error messages

Set human goals, be tolerant, consider users' feeling

Be positive, constructive, not negative, critical
Stress users' ability to control

Be as specific as possible

Write messages for audience and situation
Be brief, but as specific as possible

Playact messages for suitability
Test comprehensibility
Then edit language, wording

Test messages along with running code
Observation, iterative design

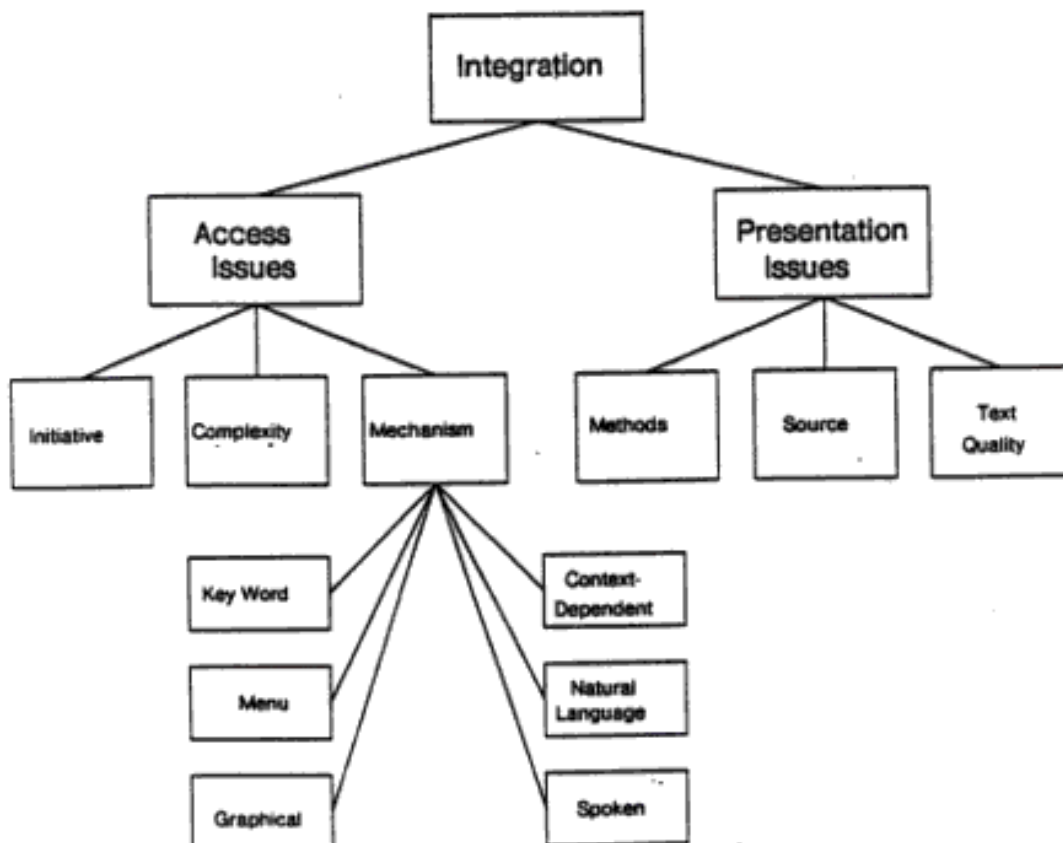
16.9 System-provided help for users

“Passive” help system not difficult to build

Must integrate with documentation

Variety of access, presentation methods (Fig. 17.1)
Importance of *text quality* – Most important factor
(Borenstein Ph.D. thesis)

Figure 16.1 Dimensions of Help System Variability
From Borenstein, N.S. (1985), *The Design & Evaluation of On-line Help Systems*,
Ph.D. thesis, Computer Science Department, Carnegie-Mellon University, p. 27



Help should answer key user questions:

Identification

What is this?

Transition

Where have I come from and gone to?

Orientation

Where am I?

Choice

What can I do now?

Demonstration

What can I do with this?

Explanation

How do I do this?

Feedback

What is happening?

History

What have I done?

Interpretation

Why did that happen?

Guidance

What should I do now?

“Active”, “intelligent” help system still a research frontier

Roles for intelligence

How to present

Literal text supplied by writers

Active examples (running code or simulations)

Explanations computed by programs

Combinations of the above

What to present

Amount of information, depth of detail

Language, level of explanation

16.10 Human help systems

Customer support from vendor

Infrastructure for help at an organization

Technological support

The role of the expert, or guru

Helping users to help themselves

16.11 Individual differences

Some key distinctions

Novices versus experts

Casual versus regular users

An over-simplification, but useful:

<i>Needs for user support</i>	Novice users	Expert users	Casual users	Regular users
Teaching and training	X		X	X
Documentation	X	X	X	X
Active learning	X	X		X
Response time		X		X
Error handling	X	X	X	X
Help	X	X	X	X

All users are different, so how to provide truly appropriate user support unsolved problem, esp. in the presence of disabilities (next lecture)

Research themes

User modelling

Interaction history

Multi-media documentation, error handling, help