What is HCI and why should we learn about it?

1. A few incidents....

2. Consequences of bad user interfaces

3. How can user interface designers determine whether a user interface will cause problems for users?

4. What is HCI?
A few incidents involving badly designed user interfaces

- A professor loses half an hour
- A professor and three staff lose two hours
- Errors are made in presidential elections
- An airliner gets shot down
- A nuclear power plant gets out of control
- ICS 205 students...
Proposal Preparation

Applicants preparing proposals for the International Opportunities for Scientists and Engineers Program Announcement (NSF 96-14) should use the 'Proposal Preparation (New)' system. The 'Proposal Preparation (New)' system contains the International Programs Cover Page Addendum.

Login for PI or Co-PI

Last Name: 

Social Security#: (9 digits)

PIN: (4 digits)

Login for Other Authorized Users

If you are not PI or Co-PI, please enter the following in addition to Last Name, Social Security# and PIN.

Proposal ID: (7 digits)

Proposal PIN: (4 digits)

If you do not know what your new PIN is or have other questions related to FastLane User Management contact your Sponsored Research Office or equivalent.
How can this problem be prevented?

Sorry..

You were able to login successfully, but you do not have permission to access the course roster for the course (ICS205) for the Spring 2000 quarter.
OFFICIAL BALLOT, GENERAL ELECTION
Palm Beach County, Florida
November 7, 2000

REPUBLICAN
George W. Bush - President
Dick Cheney - Vice President

DEMOCRATIC
Al Gore - President
Joe Lieberman - Vice President

LIBERTARIAN
Harry Browne - President
Art Olivier - Vice President

GREEN
Ralph Nader - President
Winona LaDuke - Vice President

SOCIALIST WORKERS
James Harris - President
Margaret Trove - Vice President

NATURAL LAW
John Hagelin - President
Nat Goldhaber - Vice President

REFORM
Pat Buchanan - President
Ezola Foster - Vice President

SOCIALIST
David McReynolds - President
Mary Cal Hollis - Vice President

CONSTITUTION
Howard Phillips - President
J. Curtis Frazier - Vice President

WORKERS WORLD
Monica Moorehead - President
Gloria La Riva - Vice President

WRITE-IN CANDIDATE
To vote for a write-in candidate, follow the directions on the long stub of your ballot card.

TURN PAGE TO CONTINUE VOTING
Ready for Copier access.

User Number

--XXXXX

Enter
Consequences of bad user interfaces

Bad interfaces may cause users to

• need more time for performing their tasks
• make more errors
• feel dissatisfied
• need more time for learning how to use the software
• not learn/use the full functionality of the software
• (if given a choice:) refrain from using the software
Good interface design therefore is important for any kind of interactive software, and of *utmost* importance in

- systems with high costs of failure (e.g., nuclear power plants)
- systems with high demands on operators (e.g., rescue coordination centers, combat aircraft, call centers)
- mission-critical systems (e.g., space mission control)
How can user interface designers determine whether a user interface (element) will cause problems for users?

1. Analyze the interface using "common sense" (?

   Intuitions can reveal some obvious omissions, confusing and inefficient interaction.
   Many flaws however are not "intuitively" recognizable, even with experience in user interface design.

2. Develop a theory of "human cognitive processing", and use it to predict problems that users will have with the interface

   Was only successful in limited areas so far.
3. Test the interface with users, and watch whether problems can be observed or are reported by users

Tests with 5-8 users already reveal major problems

Generalize the findings from (3) and develop guidelines of what should not be done

+ "Usability Engineering"

How can user interface designers determine whether a user interface (element) will not cause problems for users?
What is Aim of Human-Computer Interaction?

Narrow definition:

The field of Human-Computer Interaction investigates how (single) users can best interact with computers. Particular emphasis is put on

- software aspects (as opposed to the input and output devices and the physical workplace), and
specifically on the layout and operation of the interface ("User Interface Design", "Interface Engineering").

**Broad definition:**

The field of Human-Computer Interaction studies "the people side" in the interaction with computers, including

- users´ mental processes when interacting with computers
- work practices
- training issues
management of computerized work processes

- collaboration in computerized workgroups

- social/organizational aspects

- health issues

This course focusses on user interface design issues, due to their importance and the availability of other ICS courses that focus on other aspects of HCI.
Elements of HCI

1. What are user interfaces?
2. Users are different
3. Factors in HCI
4. Levels of analysis
5. Measurable human factors
6. Disciplines contributing to HCI
7. Integration of usability testing into the software development process
8. Affordances ♦
What are user interfaces?

User interfaces help users interact with programs. Users employ programs for performing their tasks.

+ User interfaces help users interact with their tasks.

+ A user interface should not reflect the structure of the underlying program, but the structure of the task domain and/or the task solution process. Users should not interact with the computer, but with their tasks.
Users are different

- Physical work environments
- Tasks
- Cognitive and perceptual abilities
- Personality differences
- Cultural differences
- Disabilities
- Age
## Factors in HCI

<table>
<thead>
<tr>
<th>ORGANIZATIONAL FACTORS</th>
<th>ENVIRONMENTAL FACTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>training, job design, politics, roles, work organization</td>
<td>noise, heating, lighting, ventilation</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HEALTH AND SAFETY FACTORS</th>
<th>cognitive processes and capabilities</th>
</tr>
</thead>
<tbody>
<tr>
<td>stress, headaches, musculo-skeletal disorders</td>
<td>THE USER</td>
</tr>
<tr>
<td></td>
<td>motivation, enjoyment, satisfaction, personality, experience level</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMFORT FACTORS</th>
<th>seating, equipment layout</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>USER INTERFACE</th>
<th>input devices, output displays, dialogue structures, use of colour, icons, commands, graphics, natural language, 3-D, user support materials, multi-media</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>TASK FACTORS</th>
<th>easy, complex, novel, task allocation, repetitive, monitoring, skills, components</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>CONSTRAINTS</th>
<th>costs, timescales, budgets, staff, equipment, building structure</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SYSTEM FUNCTIONALITY</th>
<th>hardware, software, application</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PRODUCTIVITY FACTORS</th>
<th>increase output, increase quality, decrease costs, decrease errors, decrease labour requirements, decrease production time, increase creative and innovative ideas leading to new products</th>
</tr>
</thead>
</table>
Factors in HCI.
Levels of Analysis in HCI

Figure 2.4 A model of HCI (adapted from Eason, 1991).
Central measurable human factors for user interface evaluation

- **Speed of performance**
  How long does it take to carry out the benchmark tasks?

- **Error/success rate**
  How many and what kind of errors do people make in carrying out the benchmark tasks? How many tasks were successfully completed?

- **Time to learn**
  How long does it take for users to learn what actions are required to achieve the benchmark tasks?
● Retention over time
  How well do users maintain their knowledge and skills over given periods of time?

● Subjective satisfaction
  How much did users like using various aspects of the system?
Disciplines contributing to HCI

Other contributing disciplines:
- Artificial intelligence
- Linguistics
- Ethnology
The star life cycle (adapted from Hix and Hartson, 1993).
Affordances

"term that refers to the properties of objects -- what sorts of operations and manipulations can be done to a particular object" (D. A. Norman 1988, The Psychology of everyday things).

- A door affords opening
- A chair affords support

"perceived affordance": the extent to which objects suggest their affordance.
**Figure 4.5** The affordance of objects (Gaver, 1991). (a) Door handles; (b) scroll bars (© 1991, Association for Computing Machinery, Inc. reprinted by permission).
Cognitive Elements of HCI

1. Gestalt laws
2. Graphical coding
3. Recognition versus recall
4. Mental models
5. Metaphors
6. Affordances
7. Fitts' law (see menu interaction)
8. Color vision (see color)
Gestalt Laws

Gestalt laws describe regularities of human perception (but do not explain them!)

**Proximity**: objects that are close to each other tend to be seen as a group

**Similarity**: objects of the same shape or color are seen as belonging together

**Closure**: Missing parts of an object are filled in to complete it, so that it appears as a whole.

**Continuity**: lines tend to be seen as continuous, even if they are interrupted

**Symmetry**: regions bounded by symmetrical borders tend to be perceived as coherent figures
Example for the Application of Gestalt Laws

Please answer the following questions based solely on Gestalt laws, the upper-case words in the leftmost box and the proper names in the picture.

1. How many ballots are there?

2. Can George Bush and Art Olivier together be elected for president and vice president?

3. Name all people who can be elected for president and vice president.
## Comparison of coding methods (Maguire, 1987)

<table>
<thead>
<tr>
<th>Coding method</th>
<th>Maximum number of codes</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphanumerics</td>
<td>Unlimited</td>
<td>Highly versatile. Meaning can be self-evident. Location time may be longer than for graphic code.</td>
</tr>
<tr>
<td>Shapes</td>
<td>10-20</td>
<td>Very effective if code matches object or operation represented</td>
</tr>
<tr>
<td>Color</td>
<td>4-11</td>
<td>Attractive and efficient. Excessive use confusing. Limited value for the color-blind.</td>
</tr>
<tr>
<td>Line angle</td>
<td>8-11</td>
<td>Good in special cases, for example, wind direction.</td>
</tr>
<tr>
<td>Line length</td>
<td>3-4</td>
<td>Good. Can clutter display if many codes displayed.</td>
</tr>
<tr>
<td>Line width</td>
<td>2-3</td>
<td>Good</td>
</tr>
<tr>
<td>Line style</td>
<td>5-9</td>
<td>Good</td>
</tr>
<tr>
<td>Object size</td>
<td>3-5</td>
<td>Fair. Can take up considerable space. Location time longer than for shape and color</td>
</tr>
<tr>
<td>Brightness</td>
<td>2-4</td>
<td>Can be fatiguing, especially if screen contrast is poor</td>
</tr>
<tr>
<td>Blink</td>
<td>2-4</td>
<td>Good for getting attention but should be suppressible afterwards. Annoying if overused. Limit to small fields.</td>
</tr>
<tr>
<td>Reverse video</td>
<td>No data</td>
<td>Effective for making data stand out. If large area is in reverse video, flicker is more easily perceived.</td>
</tr>
<tr>
<td>Underlining</td>
<td>No data</td>
<td>Useful but can reduce text legibility.</td>
</tr>
<tr>
<td>Combination of codes</td>
<td>Unlimited</td>
<td>Can reinforce coding but complex combinations can be confusing</td>
</tr>
</tbody>
</table>
Mental models

Mental models are representations of the function and/or structure of objects in peoples' minds.

- may be incorrect or incomplete
- can be "executed"
- are analogical representations, or a combination of analogical and propositional representations
- are dynamically constructed when required

Two main types:

- Functional models (good for everyday use)
- Structural models (good for breakdown situations; difficult to acquire from usage experience only)

* Computer systems should be designed in such a way that users can quickly acquire a good functional model of the system which is in accordance with their task model.
Important interaction metaphors

Metaphors constitute the user's initial mental model of the system's structure and operation. They should relate to users' past experiences and should be consistent.

- **(Typewriter metaphor):** Evoked easily due to physical similarities. Should be avoided

- **Desktop metaphor:** Currently the predominant metaphor.

- **Book metaphor:** For hypertext, hypertext-like online documentation.

- **Filing cabinets:** For online documentation, system settings.
• **Office metaphor:** For collections of documents

• **Library metaphor:** For large collections of documents

• **Building metaphors, city metaphors, etc.** for virtual worlds

• **Composite metaphors**
  Combine 2 or more metaphors (like office, file cabinet and desktop)

*The learning and retention of a system's functionality is considerably facilitated by meaningful and consistent metaphors.*
ICS 205

Winter 2001
City metaphor
Input and Output Devices

1. Output devices
2. Input devices
3. Requirements for input and output devices
Output devices

- Desk-bound displays (CRT, LCD, Plasma)
- Portable displays
- Displays in helmets, eye glasses, projections onto retina
- Wall-mounted displays (projections), whiteboards
- Speech Audio
- Non-Speech Audio
- Force output
- Special output devices for people with disabilities
  (Braille display, speech, etc.)
Cater to variety of possible output devices
Input devices

- Keyboards (QUERTY, Dvorak, chord, numeric)
- Cursor keys
- Mouse (1-3 keys)
- Trackballs, trackpads, joystick, 3D-mouse
- Touch screens
- Speech input
- "Graffiti", handwriting, gestures
- Data gloves, data suits, 3D trackers
- Gaze
Special input devices for people with disabilities (e.g., foot mouse, head mouse, lip readers, etc.)

+Cater to variety
Figure 2.1  A range of keyboard layouts:
(a) neither function keys nor numeric keypad (Macintosh Plus)
(b) no function keys (Macintosh SE standard keyboard)
(c) function keys and numeric keypad (IBM PC)
(d) function keys at left, cursor keypad
(e) function keys at top
Dvorak Keyboard Layout
Requirements for I/O Devices

- Must match the physiological and psychological characteristics of users, their training and their expertise (consider age, impairments, and computer skills)
- Must be appropriate for the user's tasks
- Must be suitable for the intended work environment.
Interaction Styles and HCI Guidelines

1. Interaction Styles

2. Interaction style independent design principles

3. Command interfaces

4. Menu-based interfaces

5. Direct-manipulative interaction

6. Form Fills
Interaction styles

Command entry

Difficult to learn and to retain
Can be very efficient for trained users

+ Recommended for frequent users only (and for work under time pressure)

Menus

Easy to learn and to retain
Slows experienced users down

+ Recommended for all users when complemented by menu commands
Direct manipulation
   Relatively easy to learn, easy to retain
   Restricted in scope

+ Recommended for all users when icons are labeled and when direct manipulation is complemented by other interaction style(s)

(Form fill-in)
   Limited in scope

(Natural-language interaction)
   Technical limitations
Interaction-style independent design principles

- Strive for consistency (constancy is a special type)
  - internal;
  - external with metaphor, and other software thereby conformity with user expectations

- Provide advance information

- Provide immediate feedback

- Permit easy reversal of actions (undo)

- Prevent errors; offer help in case of errors

- Reduce short-term memory load

- (Let the user be in control)

- Cater to user diversity; allow for personalization
Interaction-style independent design principles

- Provide **shortcuts** for experienced users
Command interface style

kobsa@igor% ls
ICs205-Spring00-notes/ ICS205-Spring00.html courses.html
kobsa@igor% ls ICS205-Spring00-notes
5-human-factors.htm find-problems.htm output-devices.htm
twiddler.gif
bad-ifs.htm input-devices.htm
chord-keyboard.htm metaphors.htm
contr-disciplines.htm nsf-new.jpg what-are-ifs.htm
contr-disciplines.jpg nsf-old.jpg whatishci.htm

Command interfaces

Advantages:

- Hands remain on the keyboard
- Fast input
- Powerful language constructs possible ("print file1 file2 file3", wildcards, pipes, ...)

Disadvantages:

- Difficult to learn
- Difficult to retain

Guidelines:

- Use action-oriented command names that come in the first place, i.e. <command> <arg1> <arg2> ...
  (e.g., "move a b" rather than "a move b" or "location a b")
- Use congruent names for antagonistic commands (Carroll 1982)
- Use consistent order of arguments
- Specify direct object in first argument (Barnard et al., 1982)

- Allow natural/consistent abbreviations for command names (e.g., "mv" for "move"; cf. Shneiderman 1998)

- Allow for variations in the syntax; tolerate spurious words (e.g., "mv a to b")

- Allow for the definition of aliases

- Provide command history

- Allow for the editing and re-entry of the last few commands

- If possible, allow multiple parameters and wildcards

- If possible, allow macro definition and programming
Menu-based interfaces

Overview

Guidelines for menu items

Guidelines for menu length and item order

Guidelines for menu dynamics

Guidelines for menu hierarchies

Guidelines for graphical menus

Guidelines for adaptable and adaptive menus
Menu-based Interaction -- Overview

3 basic activities:

- Navigation (in menu hierarchy, information resource, etc.)
- Selection (of data, parameters, etc.)
- Activation (of programs, documents, etc.)

Selection made through mouse, number/character keys, function keys, cursor keys plus return key, rotating cursor symbol plus return key, touching on the screen, etc.

Special case: "analog" menus

Types of menus
- Textual menus
- Graphical menus (icons)
- Combination of text and graphics
- Linear menus
- Menus with spatial layout
- Static menus (e.g., menu bar)
- Pull-down menus
- Pop-up menus
- Isolated menus
● Connected menus (mostly hierarchical menus)  

● Pie menus
Figure 1.1. Examples of the current variety of menu selection types. Portions of menus from (a) The Source™ time-sharing network, (b) a Basic interpreter on the IBM-PC, (c) the Macintosh™ Finder menu, and (d) the MacPaint® tool palette.
<table>
<thead>
<tr>
<th>Action</th>
<th>Brush</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draw</td>
<td>•</td>
<td>Black</td>
</tr>
<tr>
<td>Type</td>
<td>•</td>
<td>Red</td>
</tr>
<tr>
<td>Graph</td>
<td>⬤</td>
<td>Green</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Size</th>
<th>Face</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draw</td>
<td>9 pt</td>
<td>Normal</td>
</tr>
<tr>
<td>Type</td>
<td>10 pt</td>
<td><strong>Bold</strong></td>
</tr>
<tr>
<td>Graph</td>
<td>12 pt</td>
<td><em>Italics</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Type</th>
<th>Axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Draw</td>
<td>Bar</td>
<td>Grid</td>
</tr>
<tr>
<td>Type</td>
<td>Line</td>
<td>Tics</td>
</tr>
<tr>
<td>Graph</td>
<td></td>
<td>None</td>
</tr>
</tbody>
</table>

A simple set of cycling softkeys with commands and parameters
Pop-up pie menu
Pull-down pie menu
Guidelines for menu items

- The menu should have a short meaningful headline, preferably centered, in upper/lower case; unnecessary emphasis/embellishment should be avoided.

- Menus should show
  - the menu items that can be selected
  - the menu items that currently cannot be selected (deemphasized)
  - the menu items that have already been selected
  - the presence of submenus
  - if applicable: the code/key to select the menu item
  - if applicable: the shortcut that can be used to select the item, in addition to the mouse
Guidelines for menu items

○ If applicable: short and polite instructions on how to select items, formulated in terms of the user's task.

● Entries should be in upper/lower case; avoid unnecessary emphasis/embellishment.

● The difference of each entry to all other entries must be very clear (Schwartz & Norman, 1986).

● If the name of an item consists of more than one word, the most significant word should come first.

● If letter codes are being used, the code should correspond to the first letter of the first word. If this is not possible, numeric codes should be used (Pellman, 1984).

● To achieve external consistency with earlier versions or competitive products, menu items should be *lexically* identical (least there won't be a transfer effect; Foltz et al., 1988). Addition and omission of menu entries is o.k.
In non-tabular menus, the error rate for items with text and icons combined is 50% less than for mere textual items (Muter and Mayson, 1986).

Example of poor menu design ✿
Example of better menu design ✿
Figure 6.12. Response selection time as a function of the response-option compatibility.
Example of poorly designed menu

**MENU**

A  PROJECT ADDITION
B  PROJECT INFO UPDATE
C  DELETE RECORD
D  PROJECT INFO DISPLAY
E  REPORT
F  REPORT CHANGES

KEY:
Improved design

Update project descriptions

Options:

1. Add project
2. Update project
3. Display project
4. Delete project
5. Project report
6. Changes over previous report
7. Return to previous menu

Please enter selection
by pressing key 1 - 7: _
Improved menu design

(Type H for help)
Guidelines for menu length and item order in linear menus

- For beginners, menus should be kept short. If item names are longer than one word, menus should not contain more than 5-8 items.

- Longer menus are possible if items are grouped or have a natural sequence (numbers, dates, etc.)

- Items should be grouped according to rules of the task domain, and separated by whitespace or horizontal lines (Parkinson et al., 1985).

- For beginners: menu items that will be frequently used by most users should go to the top of the menu.

- If multiple selections are possible, then frequently-selected combinations of items should be kept closely together (or otherwise increased in size).

- "Dangerous" items should not be collocated with frequently used items.
● (As a final rule): use alphabetic, chronological or numerical order, or group according to similarity of items (e.g., position "save as" next to "save")

● Possibly use pie menus, which are superior to pull-down menus for beginners.
Selection time for menu items

The time necessary to select an item $i$ from a menu of length $n$ is composed of the search time $S$ plus

- in the case of selection via a keyboard code: the time necessary to press the key(s);
- in the case of selection with a pointing time: the positioning time $P$, plus the activation time (e.g. the mouse click).

1. **Search time $S$**

1.1 Beginning users:

- If the label of the item is unknown, all items must be inspected at least once. $S$ is therefore proportional to $n$.
- If the label of the item is known, there are three possible search strategies ◆
(1) Serial inspection
(2) Random inspection without repeat (implausible)
(3) Random inspection with repeat

In all three cases, $S$ is proportional to $n$. In case (3), both the expected means and the variance is higher than in (2) and (1).

1.2 Advanced users:

Advanced users remember the approximate position of frequently used items fairly well. The search time $S$ then is nearly constant (i.e., independent of $n$).

* The length of a menu only influences the search time when the user is a beginner.
Figure 4.3. Three models of visual search of menu lists.
2. Positioning time $P$

Fitt’s law: $P = C_1 + C_2 \log_2 \left(\frac{2D}{W}\right)$

Some authors: $+ C_3 \log_2 \left(\frac{C_4}{W}\right)$

$P = \text{Positioning time}$

$D = \text{Distance between Cursor and Object}$

$W = \text{Size of the Object}$

$C_i = \text{Constants depending on pointing device}$

---

Diagram showing $\log_2(2D/W)$ vs $2D/W$. The graph shows a logarithmic relationship.
Figure 6.3. Response time as a function of type of organization for explicit targets and for definitions.
Menu dynamics

- The menu item that is currently being touched by the cursor symbol should be highlighted.

- Submenus of the currently touched menu item should be shown (30% less search time and 50% less errors for beginning users, Snowberry et al., 1985).

- Selected options in the menu should remain checked.

- It must be possible to leave the menu without selecting any item (since users may have erroneously launched the menu, or just want to explore the system)

- Positional constancy in menu entries is important. If an item is not available at the moment, it must nevertheless remain in the menu, but should be made less prominent (e.g., grey color)

- Menus should be easily visible against all possible backgrounds.
One of the things you want to consider when remodeling your home is knowing what you need in the living space. When you consider this aspect of remodeling, get a feel for what you need and would settle for. Then outline this in a plan that you would settle for.

What aspects of interior design are pleasing to you? Would you consider the utilitarian aspects? Do you need more living space? And is it possible for you to consider the different layout of the house satisfactory? Then it is possible for you to consider the different layout of the house satisfactory.

The last command in the previous state goes to the location of cursor. However, the exact end of file location of cursor.

Inserted text to the location of cursor.
Figure 6.1. Figure-ground segmentation of the screen display. (Top panel: Good figure-ground distinction. Middle Panel: Moderate distinction with figure over figure. Bottom Panel: Poor distinction with figure over distracting figure.)
Guidelines for menu hierarchies

- Submenus of the currently touched menu item should be automatically shown (30% less search time and 50% less errors for beginning users, Snowberry et al., 1985).

- Menus should not be deeply nested (2-3 levels only).

- The menu length should not exceed 3-12 items. Menus may be longer if they are grouped or if items have a natural sequence.

- On the top level and particularly on the bottom ("leaf") level, menus should be longer than in the middle of the hierarchy (Norman & Chin, 1988).

- Longer menus are better when working under time pressure (Wallace, 1988)

- The total number of different menus in a hierarchy should be low (Norman 1991). If possible, combine two or more intermediate menus into a single one and deactivate those menu items that do not occur in the respective navigation path.
● All entries of the same level should fit on a screen (no scrolling should be necessary).

● The hierarchy should be constructed thematically. Sometimes menu hierarchies can be constructed based on which items are frequently selected together.

● Constructing hierarchies top-down or bottom-up? (Chin, 1986):
  ○ Top-down approach yields broader trees (more menu entries on the top level) and deeper trees. The emphasis lies on the discrimination of the differences between items.
  ○ Bottom-up approach yields narrower trees with broad lowermost level. The emphasis lies on the clustering of related items.
    * Use a combined approach.

● Consider menu shortcuts to deeper hierarchy levels.

● External consistency (with previous program versions or competitive programs): changing the hierarchical position of menu entries is o.k., as long as lexical identity is preserved (Foltz et al., 1988).
Guidelines for graphical menus

- Menu items (= "icons") should be recognizable and distinguishable within the normal viewing distance, under different light conditions.
- Should be esthetically pleasing.
- Global properties (like form, color or size) are much more important than local properties.
  - Abstract icons become much faster selected than concrete icons.
  - The selection speed for abstract items seems to be less dependent on the number of icons than is the case for concrete and textual items ("parallel search").
- Similar objects/functions should be assigned similar icons.
- Users should be able to easily understand the intended meaning of icons, or at least be able to learn it quickly.
Figure 6.9. Word command menu, distinctive/abstract icon menu, and representational icon menu used by Arend, Muthig, and Wandmacher (1987).
Figure 6.10. Mean response times to word commands, distinctive icons, and representational icons as a function of menu size.
Guidelines for adaptable and adaptive menus

- Experienced users (and system administrators) should have the possibility to make changes to the menus, to cater to user needs (= "adaptable menus"):
  - introduce/change shortcut codes
  - hide/delete redundant menu items
  - move/duplicate menu items into other menus
  - create links between menus

- It does not seem advisable to allow for completely automatic positional changes in menus, like re-ordering or hierarchical re-positioning of menu items based on usage frequency (= "adaptive menus")
  - violates the constancy principle
negative experimental results by Shneiderman

- It seems o.k. to perform positional changes *under the control of the user.*

Recommendations for changes in a Word 97 "Personal Toolbar", based on usage frequency (Spasovsky/Fink/Kobsa 1997)
Direct-interactive style
Form-fills

- Useful for entering large amounts of data
- Consist of tables and label/data fields

Guidelines

- Switch at most once between tables and label/data fields.
- Left-align labels, data fields, and columns of tables.
- Alphabetical listings should be arranged vertically, possibly in several parallel columns.
- If user enters data from a paper document, its layout should be mirrored by the screen layout.
- Allow for a space of at least two characters between columns.
● Insert a blank line after about every fifth line of data.

● Split long alphanumeric codes in groups of 2-4 characters.

● If a specific format of the user input is required, communicate this to the user:
  ○ Use underscores, periods etc. to indicate the exact or maximum input length.
  ○ Units (like $, ft., mph) and separators (like "-", "/") should be part of the form.
  ○ Use add-ons like "dd/mm/yy" or "(example: 12/_8/59)".

● Mark fields where entry is required (e.g., by "?", "." or color), or group such fields separately. Hitting the return or tab key should bring the user to the next required entry field.

● Fill in data fields with default values or with the most recently entered value.

● Enter field values automatically if they can be inferred from other user input.
Allow for abbreviated entries which can be expanded by hitting a single key (automatic expansion is difficult for beginners). If the result is not unique, show available alternatives.

Automatically entered values should however never have dangerous effects.

Example
Field Organization

Our eyes naturally or culturally move from top to bottom, left to right. Place fields to take advantage of this standard, and move the cursor accordingly. For example, arrange related fields across the screen as in

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Task:</th>
<th>Duration (Days):</th>
</tr>
</thead>
</table>

or down it, as in

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Task:</th>
<th>Duration (Days):</th>
</tr>
</thead>
</table>

but not in a zigzag pattern:

<table>
<thead>
<tr>
<th>Resource:</th>
<th>Task:</th>
<th>Duration (Days):</th>
</tr>
</thead>
</table>

Use columns for repetitive information, since they complement natural eye movement. Note how part number, quantity, price, and extended price flow from left to right in

<table>
<thead>
<tr>
<th>PART NO</th>
<th>QTY</th>
<th>PRICE</th>
<th>EXTENDED</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

rather than

<table>
<thead>
<tr>
<th>PART NO</th>
<th>(1)</th>
<th>(2)</th>
<th>(3)</th>
<th>(4)</th>
<th>(5)</th>
</tr>
</thead>
</table>
An exception occurs with paired groups of fields. For instance, if your application has “sold to” and “ship to” address information, arrange the fields as

<table>
<thead>
<tr>
<th>Sold To</th>
<th>Ship To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name:</td>
<td>Name:</td>
</tr>
<tr>
<td>Addr1:</td>
<td>Addr1:</td>
</tr>
<tr>
<td>Addr2:</td>
<td>Addr2:</td>
</tr>
<tr>
<td>City:</td>
<td>City:</td>
</tr>
<tr>
<td>St:</td>
<td>St:</td>
</tr>
<tr>
<td>Zip:</td>
<td>Zip:</td>
</tr>
</tbody>
</table>
Organize lists vertically like this:

Adrian  Charles  Fatima
Agrippa  Glenda  Florence
Basil  Dana  Gordon
Beverly  Douglas  Hanna
Caine  Edith  Henry
Cary  Eliot  Hope

Rather than horizontally like this:

Adrian  Agrippa  Basil
Beverly  Caine  Cary
Charles  Glenda  Dana
Douglas  Edith  Eliot
Fatima  Florence  Gordon
Hanna

Henry

Hope
**Poor:**

<table>
<thead>
<tr>
<th>Name:</th>
<th>Major:</th>
<th>Year:</th>
<th>Status:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID:</td>
<td>Co-op Track:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Section</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Improved:**

**STUDENT REGISTRATION**

**STUDENT**

<table>
<thead>
<tr>
<th>Name:</th>
<th><em>Major:</em></th>
<th>Year:</th>
<th>Status:</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID:</td>
<td>Track:</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COURSES**

<table>
<thead>
<tr>
<th>Number</th>
<th><em>Title</em></th>
<th>Section</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Figure 4-12. Using dots to indicate field length.

**JONAH'S FISHING EXPEDITIONS**

**NEW EMPLOYEE ENTRY**

<table>
<thead>
<tr>
<th>7/1/99</th>
<th>SCREEN</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:15 am</td>
<td>1 OF 2</td>
</tr>
</tbody>
</table>

**REQUIRED DATA**

- **EMPLOYEE NO:** ............
- **SSN:** ............
- **EMPLOYEE NAME:** ............
- **ADDRESS1:** ............
- **ADDRESS2:** ............
- **CITY:** ............
- **ST:** ............
- **ZIP:** ............
- **DIVISION Code:** ....
- **dept code:** ....
- **section code:** ....

**OPTIONAL DATA**

- **SEX (M/F):** ............
- **BIRTHDATE (MM/DD/YY):** ............
- **SALARY GRADE:** ............
- **MONTHLY SALARY:** $ ............

**F1-Help  F2-Next Screen  F3-Add Employee  PgDn-Benefit Plans  Esc-Cancel**
Figure 4-13. Using dots with numeric data.

Figure 4-14. Using colors to indicate field length.
Figure 4-9. Filling fields to show required entries.
Figure 4-11. Separating required data from optional data.
| VIDEOTAPE NO: | _______ |
| MOVIE TITLE: | Cas______ |
| QTY: | ______ |

F1-Help  F3-Update Quantity  Esc-Cancel

**Figure 4-18. Finding an entry when the complete title is not known.**
QTY: 2
Enter quantity or press <PgDn> for next title

F1-Help    F3-Update Quantity    Esc-Cancel

Figure 4-19. Showing the first record matching user input.
Figure 4-20. Popup window showing possible choices.
Figure 3-14. Initial screen design for Famous Surgeons' Mailing List Program.

Figure 3-15. Improved screen design for Famous Surgeons' Mailing List Program.

Figure 4-25. Revised screen design for Famous Surgeons' Mailing List Program.

Figure 4-26. Alternative screen design for Famous Surgeons' Mailing List Program.
Selected elements of screen designs

1. General screen layout
Guidelines for general screen layout

- The layout should reflect the structure of the task or the task solution process, and not the structure of the underlying program.

- All information that is necessary to solve a coherent sub-task should be visible on the same screen.

- The screen should not contain information that is never relevant for the user.

- The screen should contain a minimum of 60% (and ideally 85%) whitespace.

- The upper and the lower half of the screen should roughly contain the same amount of whitespace.

- The screen layout should be more or less vertically symmetric.

- All information that belongs together should be grouped together in clearly visually separated unit that is always presented at the same place.
• Proximity/distance is mostly good enough for grouping;
  • if not, use lines, differently colored background, or boxes.

• Users should be able to enter and correct data in arbitrary order.

• If connecting screens have to be used,
  • the same headlines should be used
  • information units needed in two or more screens should be presented on all screens, at the same location.

• Users should always be able to find information on
  • how to get to the previous screen
  • optional: how to exit the subtask or the whole program
  • how to obtain help

• Unnecessary colors and embellishments should be avoided.

• Emphasis should only be used if really necessary, and only with necessary prominence:
- **low:** different font, underlining, upper case, spacing out
- **medium:** different size, color or density, inverse video, framing
- **high:** blinking, acoustic signals

**Special screen areas**

**Title line**
- centered in the upper part of the screen
- separated by whitespace
- possibly marked out by larger font, bold face, inverse

**Lowermost screen part:**
- Good place for status information
- explanation of available secondary menu options (help, exit, special function keys)
- dynamic explanation of items that are touched by the mouse symbol
- warnings and error messages (possibly with an acoustic signal that can optionally be turned off)
Logos: should go into the left or right upper corner
Clocks: should not show seconds and should not tick

Example
Figure 3-4. Screen with poorly used whitespace.
Figure 3-5. Screen with better use of whitespace.
Figure 3-7. Unbalanced screen.
Figure 3-8. Screen with improved balance.