XXI. User Interface Design

What is Human-Computer Interaction?
Affordances, Mappings, Mental Models,
Feedback, Forcing Functions, Learning
How to Design Interfaces
User Dialogue Design
Inputs and Outputs

Why Should a Systems Analyst Know HCI?
- 40-60% of today's software consists of user interfaces.
- Such interfaces support high interactivity with the user, much end-user programming
- User interactivity is only going to get more complex: 3D graphics and virtual reality, augmented reality activities....
- Many usability problems due to a bad interface design:
  - Users sabotage systems they don't understand;
  - Users make more errors when dealing with systems that are difficult to use.
- A well-designed user interface can reduce coding costs, interface problems, life-threatening errors; can also lead to increased sales, ....more...

Increased Productivity

20 users
X 230 days
X 100 screens per day
X 10 sec per screen (savings)
= 1278 hours or 32 weeks

Reduced Training Costs

20 employees
X 2 systems/applications per year
X 2 1/2 days per application (saved)
= 100 days or 20 weeks of savings

Preventable User Errors

500 users
X 20 errors per year
X 15 minutes per error
= 2500 hours lost or 63 weeks
### Increased Productivity

- **500 menu selections per day**
- **2 sec per selection (saved)**
- **250 days per year**

\[ \text{For a $100K person-year cost, saved time translates to $15,000} \]

### Serious Life-Threatening Errors

- Analysis of transcript of 911 call announcing bomb in Centennial Park at the Atlanta Olympics indicated that 20' were needed to call dispatchers
  - Dispatch system required an address; operators could not find anyone who knew address;
  - Bomb was set to go off 30 minutes after call.
- Airline crashed into a mountainside in Colombia in 1996, killing all aboard
  - Pilot typed in “R” rather than full name of airport;
  - Guidance system selected the first airport in the list beginning with “R” -- which was the wrong airport;

### User Interface Economics

- Good user interfaces sell systems!
  - Windows is a copy of the Macintosh interface;
  - The Mac interface is a copy of Bravo - developed by user interface researchers at Xerox PARC.
- User interface capabilities and awareness help get contracts.
- Poor user interfaces can cripple a system that is outstanding in all other respects.
- Poor computer-driven interfaces placed in most mechanical products we know
  - Who can set the clock on their VCR?
  - Who can use photocoppy, fax, candy, bank machine, cash register, telephone…;

### Why Are User Interfaces Poor?

- Inadequate training of people developing interfaces.
- Diverse knowledge required to design interfaces.
- Rapid technological advances.
- Poor management - programmers do not talk to user design teams and vice versa.
- User Interface specialists rarely involved.
- The “bricklayers” (programmers) are left to design the user interface, by default.

> "Ignorance by software engineers of usability and how to measure it is roughly equivalent to an electronics engineer not knowing what volts and watts are and how to measure them."

### What’s Wrong with this Interface?

#### Design of User Interface for CAVECAT Media Space

#### Why is this Design Better?

#### Design of User Interface for Telepresence Media Space
Some Basic Human Characteristics

- Humans like problem solving, for solvable problems!
- Humans are always learning, but learning is hard!
- Humans use prior learning to support new learning.
- Users don’t read manuals, work by copying and asking.
- Users are always building models of their world.
- Implications
  - Build interfaces that allow people to learn by using the interface;
  - Build interfaces that suggest correct models;
  - Build interfaces that rely on prior learning.

More Human Characteristics

- Users don’t mind if something doesn’t make sense - they build their own model to make it make sense.
- Users prefer simple models.
- Inconsistency doesn’t bother users -- A simple model which doesn’t always match is better than a complex model that is too hard to learn.

Features of a Good Design

- Has affordances - makes each operation visible;
- Offers obvious mappings - makes the relationship between the actual action of the device and the action of the user obvious;
- Provides feedback on the user’s action;
- Provides a good mental model of the underlying behaviour of the device;
- Provides forcing functions -- prevents a user from making bad errors;
- Supports automatic learning -- offers consistencies and practice that help the user acquire interface skills.

Affordances

- The design of the artifact in some way describes what the user can do with it, i.e., its affordances.
- Good example of affordance -- buttons which indicate to the user that they are to be pressed.
- Bad example of affordance -- the “put-away” box in the upper right hand corner of a Macintosh window.
- Well-known affordances:
  - Glass is for looking through (...or breaking)
  - Stairs are for climbing...
  - Cardboard is for writing on...
  - Radio buttons are for pushing or turning...
  - Door handles are for pulling, door bars are for pushing...

Different Affordances

Mapping Functions

- Mapping Functionality - the design in some way shows a mapping between the intended use of the tool and the action of the underlying technology.
- Good example of mapping: -- the presentation of the font to be selected in the form and shape of the font.
- Bad example of mapping: function keys in general -- the mapping is totally arbitrary.
Some Designs are Better Than Others...

Paired Stove Controls

Fully Natural Mapping of Controls and Burners

Mental Models

- Mental Model: understanding a person has about how technology or device works, so that user has some idea that if she performs action A, then event B will follow.
- Examples of incorrect mental models:
  - Some foreign students apply directly to a professor for graduate studies;
  - Some parents of foreign students try to find a friend within the university who will influence the admissions office.
- Good example of mental model usage: WYSIWYG - What you see is what you get.
- Example of systems with no mental model: online retrieval systems, extra functions on a telephone

Forcing Functions

- Forcing Functions: are designs that prevent users from taking actions which are inappropriate or which would lead to error.
- Good example of a forcing function design: the Macintosh menu bar - grays out and prevents selection of items inapplicable to the current context
- A bad example of a forcing function design: Unix - every command is allowed, if typed correctly
- Exercise: You buy some groceries on your way to work and put them in the office refrigerator; how do you make sure that you won’t leave work without your groceries?
Feedback

- Feedback - a design in which a form of visual, auditory or other modality response is given immediately after the user action to indicate that the action has been received.
- Good example of feedback: icons on the screen which show a reverse video image when selected.
- Example of non-use of feedback: Latex and like-minded text formatting systems.

Automatic Learning

- A design can force learning by offering repetitive patterns of user actions or screen displays.
- Good feature of automatic learning: user actions always involve same number of steps, e.g., select object, select general action to perform on object, select specific case of action.
- Example of non-use of automatic learning: Screens which change standard menu item locations from display to display.
- Bad example of automatic learning: A confirmation action that always requires a carriage return.
- Good example of automatic learning: Confirmations that require some knowledge of context, e.g., the first character of the file to be deleted.

Where are the Affordances, Mappings, etc.?  

- Affordances: Slider on the right side. Arrows at the top and bottom suggest sliding the bar (even though they are buttons). The size/location of the bar suggests the allowable directions that this can be slid.
- Mappings: Search button, the icon with the magnifying glass. Magnifying glass used to look for things/expand things.
- Feedback: The scrolling logo on the top right to indicate that a search for a page is in progress. Tells the user that their last jump to a hyperlink is being processed.
Mental Model, Forcing Functions...

- Mental Model: Following a sequence of links forms a chain. The UI allows navigation of this chain via the forward and back buttons.
- Forcing Functions: The forward/back buttons are enabled only if navigation of the chain in the specified direction is allowed.
- Automatic Learning: Links always consistently highlighted, visited links consistently highlighted. OR, interface uses the Netscape/Explorer layout and functions.

Are We Good Designers?

- Do we constantly bump into things, knock our head, hurt our knees etc.? Do we avoid moving the furniture so that it creates a forcing function that prevents us from walking into something?
- Do we store things with no identification labels that would provide a mapping function to the item we want, e.g., keys on a ring that all look alike?
- Do we respond to email confirming that a time has been set and the message has been received, thus giving feedback to our friends?

Designing User Interfaces

- I/O Design: Decide who inputs what data when:
  - Batch input/output, e.g., read data from file to update database at 7pm daily, produce a report each Friday;
  - Interactive input and/or output, such as customer access their accounts at the rate of 1,500/hr
- Dialogue Design: For each input and/or output session design the dialogue structure that will be supported; for example, an ATM session dialogue structure involves user inserting card, system prompting for PIN etc.
- Screen Layout and I/O Format Design: For each interactive dialogue, design screens that will be presented to the user; for each batch I/O design the format of the input data, or the output report.

User Groups

- In general, an information system will be used by several different groups, including non-technical people (clerks, managers) and technical people (system operators, database administrators, ...)
- Each one of these groups may require its own interface (some assuming no technical background on the user’s part, others assuming a lot)
- End users are the non-technical users of an information system.

User Interface Medium: Monitors

- Monitors used to display input/output; key characteristics of monitors:
  - Display area -- how large is the screen;
  - Character sets and graphics -- older monitor technology was character-based (i.e., the monitor could display one of X characters in one of N screen positions, e.g., 80x60); new technology is bitmap-based (i.e., monitor can display a point of different grayscale intensity/colour in one of N screen positions, e.g., 80x640);
  - Paging and scrolling -- data are displayed page-at-a-time, or continuously through scrolling
Windows and Graphical User Interfaces (GUIs)

- Windows provide a user-defined partition of the screen into multiple working areas.
- Windows have become an interface standard, with OSF Motif (Unix) Microsoft Windows, Apple MacOS.
- Graphical user interfaces (GUIs) use icons (graphic symbols), pop-up windows, scroll bars and pull-down menus; also radio buttons, check boxes and dialogue boxes.
- User friendliness is enhanced by a mouse, trackball, pen or other pointing and input device which reduces the need for a keyboard.

Layout Concepts

- The screen is often divided into three boxes: Navigation area (top), status area (bottom), work area (middle).
- Information can be presented in multiple areas.
- Like areas should be grouped together.
- Areas and information should minimize user movement from one to another.
- Ideally, areas will remain consistent in size, shape, placement for entering data, reports presenting retrieved data.

Content Awareness and Aesthetics

- All interfaces should have titles.
- Menus should show clearly where you are, also where you came from to get there.
- Should be clear what information is within each area.
- Fields and field labels should be selected carefully.
- Use dates and version numbers to aid system users.
- Interfaces need to be functional and inviting to use.
- Don’t squeeze in too much.
- Design text carefully, be aware of font and size, avoid using all capital letters.
- Colors and patterns should be used carefully, test color quality by trying interface on a black/white monitor, use colors to separate or categorize items.

User Experience and Consistency

- How easy is the interface to learn?
- How easy is the interface to use for the expert?
- Consider adding shortcuts for the expert.
- Where there is low employee turnover, some training can lessen the impact of less precise interfaces.
- Consistency enables users to predict what will happen and reduces learning curve.
- Consistency concerns items within an application and across applications.
- Consistency pertains to many different levels:
  - Navigational controls;
  - Terminology;
  - Report and form design.

Dialogue Modes

- **Menu selection** -- user given a number of options listed on a menu, selects one and the system carries out the option selected or updates its database accordingly, then displays another menu; e.g., MacOS and applications.
- **Instruction sets** -- dialogue structured around instruction sets which provide the user with a command language (using structured English, mnemonics or free-format syntax; e.g., Unix).
- **Question-Answer dialogue** -- system-driven (as opposed to user-driven) Q-A easier.
- **Graphics-based dialogue structure** -- monitor+mouse capabilities; uses menus but also many other features.

GUIs clearly the way of the future
**Designing Dialogue Structure**

One can use state diagrams to specify dialogue structure to be supported by a user interface. Such diagrams usually have only input from the user (“event”), and output (“action”) to the user.

- **Card entered**/
  - PIN?/
    - **Valid PIN**/
      - **Transation?**/
        - **Withdrawal**/
          - Amount?/
            - **Deposit**/
              - Amount+Envelop?/
                - **Print receipt**
              - Amount/
                - Sufficient funds?/
                  - **No**/
                    - **Amount?**
                  - **Yes**/
                    - **Cash+receipt**
                - **Cancel**/
                  - **Return card**
          - **Balance**/
            - **Print balance**
    - **Cancel**/
      - **Return card**
        - **Yes**/
          - **Transaction?**
            - **-/Another transaction?**

**Conditional Transitions**

Sometimes transitions are conditional...

- **ATM Ready**
  - withdraw(amount)
  - generateCash(amount)

**Airline Reservation Dialogue**

- **Reservation Request**
  - Have Details on Reservation
  - Details on Cancellation
  - Details on Change
  - Cancellation Request
  - Details on Request
  - Details on Info Request

**Dialogue Structure for an ATM**

**How to Design a User Interface:**

**Iterative Design**

- **Plan** -- what needs to be learned? Identify highest risk items, user(s), tasks to be performed through the interface, time allowed
  - e.g., temporary secretary, type letters, 1hr
- **Prototype** -- prototype only pieces that are needed; intended to minimize effort and maximize feedback
- **Measure** -- gather data on how effective the prototype is; look for false, missing, useful affordances; check for overhead in learning the interfaces as opposed to doing useful work
- **Learn** -- evaluate the data to identify areas that need further elaboration

**User must be in the loop!**
**Prototyping**

- Build a "quick-and-dirty" implementation of the interface in a very high level language (Lisp, Prolog, 4GL) or GUI tools, to show users what the interface looks like.
- Do a paper mock-up using cardboard, index cards, colour markers, tape, scissors...
  - Use cardboard rectangles, flip charts to represent screen; use index cards for drop down menus.
  - Avoid technical terms, "very intelligent" help, unimplementable features.

**Paper Mock-Ups**

- Designer plays "the computer", writes on tape or transparency computer's response
- Users use their fingers as a mouse, use "typed" input on removable tape or transparency
- Mock-ups take away the intimidation of the "technology barrier", make users feel at ease; users' imagination fills the gaps
- Mock-ups can be changed very quickly (quick feedback important to users)
- Mock-ups offer only approximate look-and-feel, can't be used to assess response times

_Do users and organizations accept mock-ups? Yes, they do._

**Other Input/Output Design**

- Apart from user interfaces, through which the users input/output directly information into/out of an information system, other input or output modes may have to be designed as well.
- For example, a government information system may require a data entry interface, where staff input data read in from forms filled out by people.
- Or, output report format may be designed for bank executives who don't have the time to learn to use a particular system, but do need certain statistics.
- Below we list some of the options in designing such I/O interfaces.

**Output Design: Types of Outputs**

- **External outputs** -- leave the system permanently; e.g., paycheques, airline tickets, boarding passes,...
- **Turnaround outputs** -- leave and later re-enter the system; e.g., invoices, purchase orders
- **Internal outputs** -- never leave the system (useful for monitoring and management purposes); e.g., internal reports, summary reports etc., used for system administration

**Input Design**

- **Data capture** involves the identification of new data to be inserted in an information system, e.g., a photo
- **Data entry** is the process of translating the source document into a machine readable form
  - e.g., digitizing the photo
- **Data Input** is the actual entry of data (already in machine-readable form) into the computer

**Input/Output Media and Formats**

- An input/output **medium** is the material used to record information
  - e.g., punched cards, tape, diskette, paper or video display device
- An input/output **format** determines the way information is organized on the medium
  - e.g., for output, tables, bar or pie charts,...
Output

- **Output media** -- paper used most frequently, but also microfiche (good for archiving), video (fast-growing use)
- **Output formats** -- tabular format, zoned or form-based format places text or numbers in designated areas, graphic format uses graphs or charts to display information, narrative format uses narrative form to present information

Preparing Graphs and Charts

- **Monitor number of inputs** -- keep track of batch#, # of documents, # of lines; these can be compared with outputs
- **Data validation for inputs** -- this involves programs which check for typos, missing or suspicious information.
- **Internal controls for outputs** include specifying exactly the time, volume and destination of each output, also access controls (e.g., password).

**Data errors can be reduced dramatically because of data validation procedures**

Summary

- The acceptance of an information system by its users depends critically on its user interfaces.
- User interfaces are best developed through prototyping, involving prototype implementations or paper mock-ups.
- In designing a user interface, one needs to keep in mind cognitive principles about human information processing, and memory organization.
- Human-Computer Interaction is the area of Computer Science which studies such principles and offers ways they can be exploited to build better user interfaces.
**Additional Reading**
