Notes on
Human-Computer Interaction

Topic T1
Introduction to Human-Computer Interaction

Humans:
- Have various ways of perceiving the world, viz. seeing, hearing, touching, smelling and tasting
- Have memories, experiences, skills, knowledge, emotions, etc.
- May be children, senior citizens, differently-abled, not tech-savvy, etc.

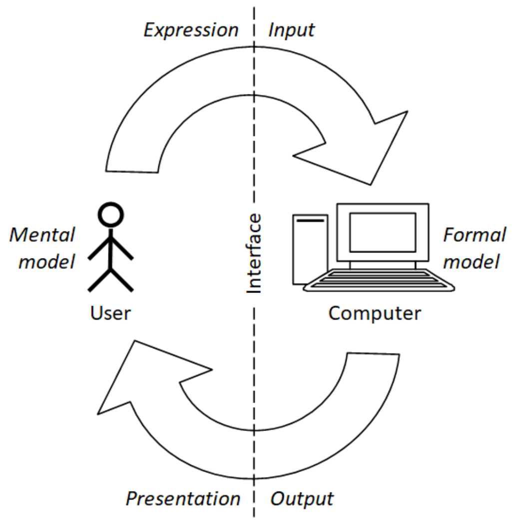
Computers:
- Machines that can implement instruction cycle
- Machines that can execute programs
- Can be of various types, viz. personal computers, smartphones and wearable devices

Smartphones:
- Limited screen area
- Imprecise input methods
- Distracted users

Wearable devices
E.g. smart watches, smart glasses, exercise monitors, prosthetic limbs, etc.

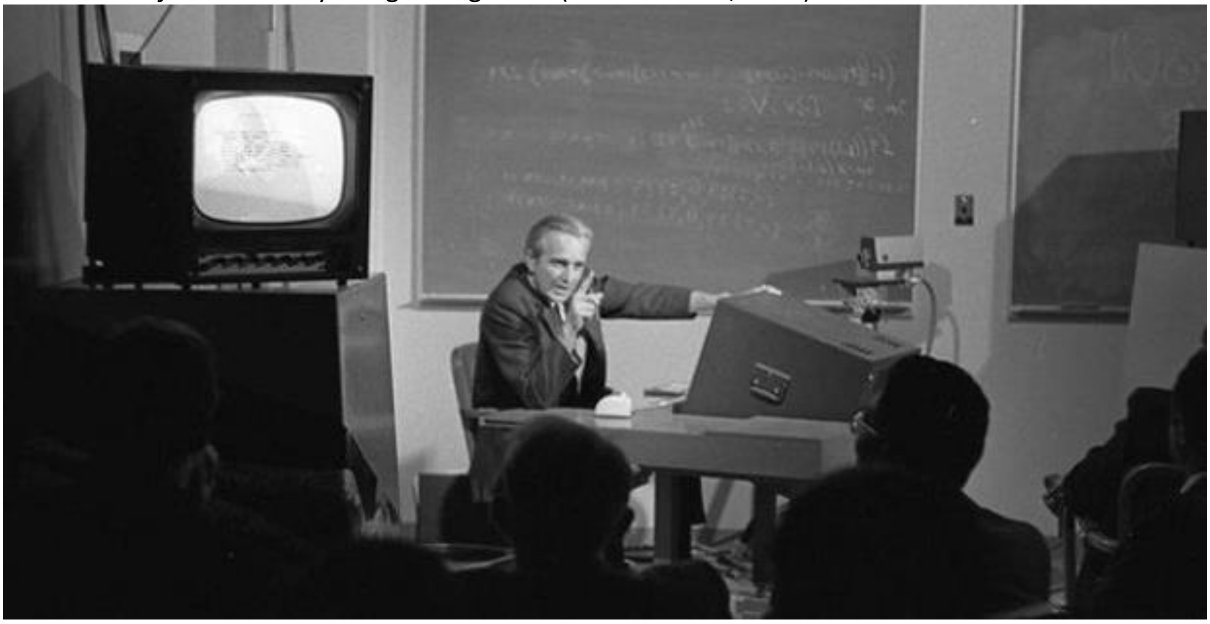
Interaction between humans and computers:
- A human being interacts with a task through a computer
- The computer mediates the interaction between the human being and the task
- The interface should be as invisible as possible
- Ideally, the interface should vanish

Multimedia = use of multiple content formats
Multimodal interaction = use of multiple modes of input/output

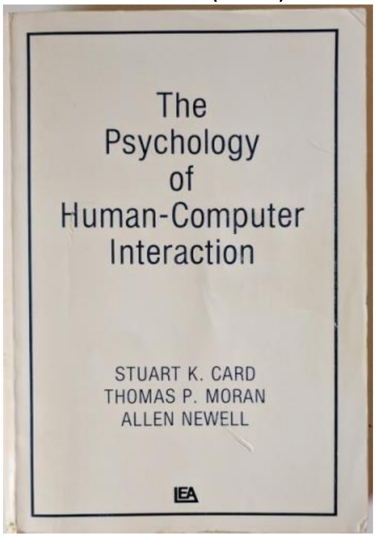


The term human-computer interaction was first used by James H. Carlisle (1975).

The Mother of All Demos by Douglas Engelbart (San Francisco, 1968):

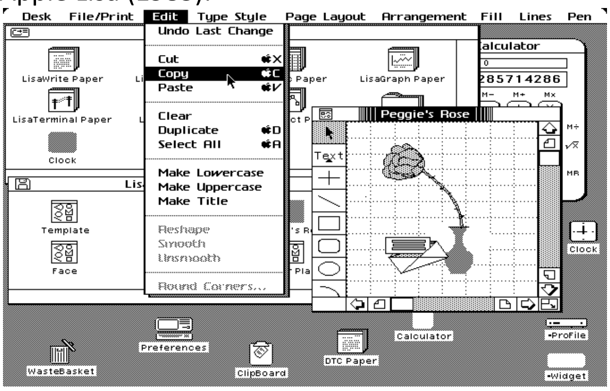


First book on HCI (1983):



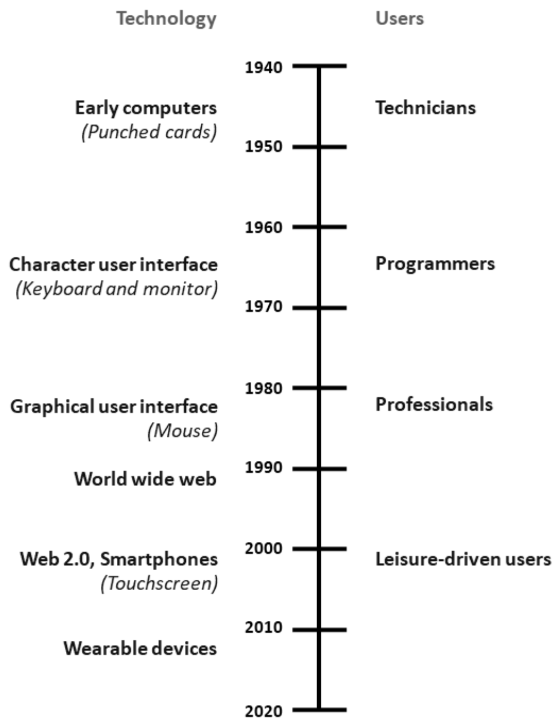
Xerox Star (1981):

Apple Lisa (1983):



Microsoft Windows (1985):

Evolution of technology and users:



Thinking about how to use the software to solve a task is frustrating for a human being. The worst way to design software is to make human beings to think how to use it.

Software engineering → Developing reliable software
Human-computer interaction (HCI) → Developing easy to use software

Goal of HCI: To design effective interactions between humans and computers

Developer ≠ User
HCI is about designing interaction of other human beings with computer.

HCI = User interface (UI) design + User experience (UX) design

HCI $\subset$ Human factors engineering
More and more devices are being computerized.
As computers are becoming more ubiquitous, the gap between HCI and human factors engineering is decreasing.

HCI is influenced by:
- Computer architecture, programming, graphics and multimedia
- Physiology, psychology, art and design
- Print technology

HCI involves:
- Studying users
- Designing systems

Software has to be usable to be useful.

Chapter 1
Usability of Interactive Systems

Designing a product to be simple and clear takes time.

The interdisciplinary design science of human-computer interaction began by combining (a) the data-gathering methods and intellectual framework of experimental psychology and (b) the powerful and widely used tools developed from computer science.

Contributions also came from graphic designers and ergonomics experts.

Human performance and user experience with computer and information systems will remain a rapidly expanding research and development topic in the coming decades.

User interfaces change many people's lives.
A good user interface can help users, while a bad user interface can frustrate users.

Successful designers go beyond the vague notion of "userfriendliness", probing deeper than simply making a checklist of subjective guidelines.
They have a thorough understanding of the diverse community of users and the tasks that must be accomplished.

Effective interfaces generate positive feelings of success, competence, mastery and clarity in the user community.
The users are not encumbered by the interface and can predict what will happen in response to each of their actions.
When an interactive system is well designed, the interface almost disappears, enabling users to concentrate on their work, exploration or pleasure.
Creating an environment in which tasks are carried out almost effortlessly and users are "in the flow" requires a great deal of hard work by the designer.

Try to increase performance, reliability and standardization.
Try to decrease personnel requirement, skill requirement and training time.

Standardization refers to common user-interface features across multiple applications.
Consistency refers to common action sequences, terms, units, layouts, colors, typography and so on within an application program.
Integration refers to commonalities across application packages and software tools.
Portability refers to the potential to convert data and to share user interfaces across multiple software and hardware environments.

Usability measures:

- Time to learn
- Speed of performance
- Rate of errors by users
- Retention over time
- Subjective satisfaction

The enormous interest in interface usability arises from the growing recognition (a) of how poorly designed many current interfaces are and (b) of the benefits elegant interfaces bring to users.

Reflect on the usability measures for the following types of system:

- Life critical systems
- Business systems
- Personal systems

Factors affecting usability:

- Physical abilities and physical workspaces
- Cognitive and perceptual abilities
- Personality differences
- Intercultural and international diversity
- Differently-abled users, adult users and children
- Hardware and software diversity

Goals of HCI course:

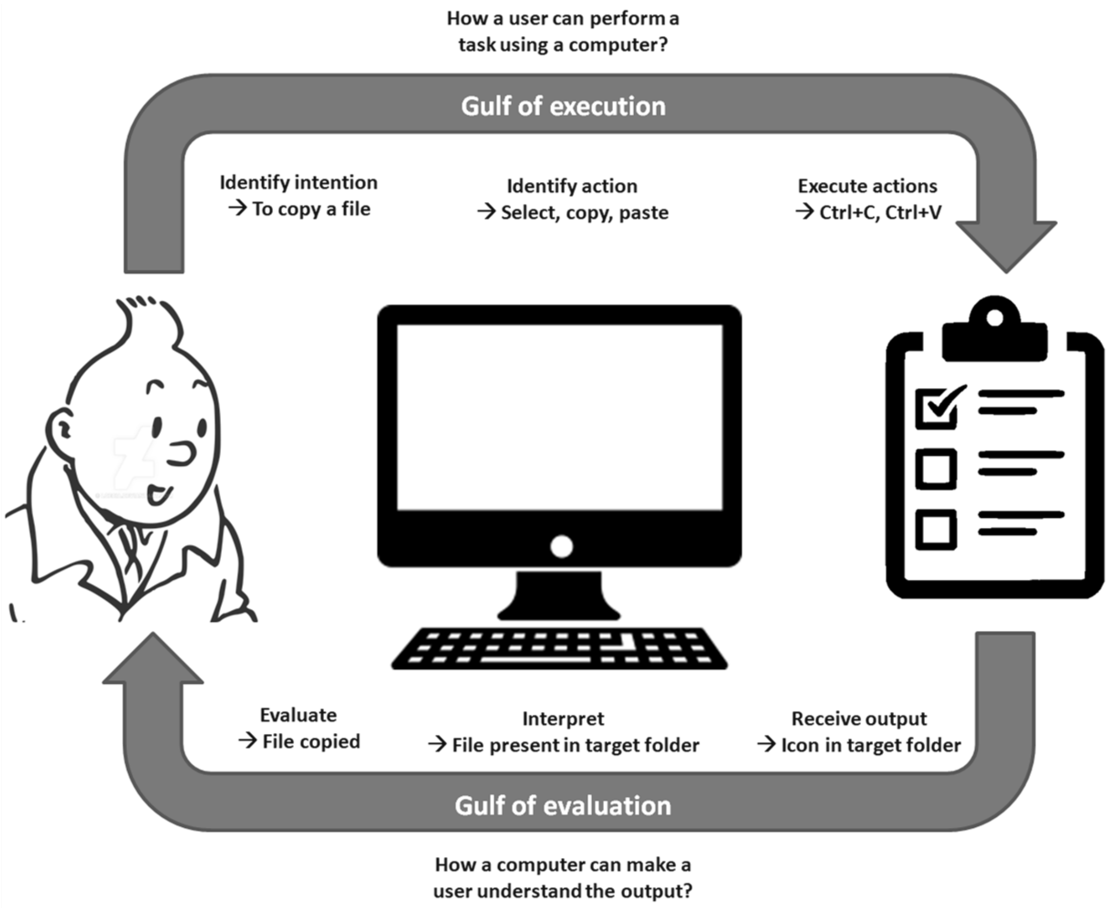
- Influencing academic and researchers
- Providing tools, techniques and knowledge to developers
- Raising computer consciousness among people

Topic T2
Designing an Interface

Human beings use tools to accomplish tasks.
The tool is a computer in our case.

To design a good interaction, we need to understand:
- The goals of the user
- The details of the task

Identifying the task:
- Watch the users
- Talk to the users
- Recognize the task bottom-up
- Different users behave differently



Reducing the gulf of execution:
- Make functions discoverable
- Let user mess around and redo
- Be consistent within the tool and with other tools
- Provide feed-forward

Reducing the gulf of evaluation:
- Give feedback constantly
- Give feedback immediately

- Feedback should be according to user action
- Vary type of feedback

Advances in hardware and software technologies have helped in reducing the gulfs:
- Graphical user interface (GUI)
- Touchscreen
- Google voice search with its audio response

UX design is about building a system that dictates how users will feel while using it.

Symbiotic relationship between software and society:
- ‘Other’ option in the gender field
- ‘Civil union’ option in the marital status field

Character user interface (CUI) → commands
Graphical user interface (GUI) → icons and menus

Commands:
- Remember commands
- Remember options
- Less feedback
- Nothing natural in this

Icons:
- Immediate visual and auditory feedback
- Animation often used for feedback

Invisible interface:
- Invisible by learning
- Invisible by design: Invisible from first use

Different meaning of invisibility for different users:
- Menu items
- Shortcut keys

Talk to users: if they talk about the interface, then it is pretty much visible.

Human abilities: UI and UX should be designed according to the abilities of the target users.

Eyes:
- Most concepts of HCI are related to visual perception
- Colors, details and movement can be detected at the center of vision
- Movement can also be detected with peripheral vision
- Colorblindness: 1 in 12 male, 1 in 200 female
- UI should not depend on colors only
- Visual acuity decreases with age

Ears:
- Hearing depends on pitch and loudness
- Ears can localize the source of sound

- Hearing is not directional and cannot be avoided, hence suitable for designing alerts

Memory:
- Three types of memory, *viz*. perception store / working memory (<1s), short-term memory and long-term memory
- Expertise can delay decay of working memory
- Recognizing is easier than to recall
- User should be allowed to control the pace, e.g. menus should not disappear
- Information entered or displayed in a previous screen may be displayed in a screen if the user needs the same

Cognition:
- How much attention user provide to the interface in real world?

Motor:
- Precision of taps and clicks should be considered, e.g. close buttons should not be placed near other buttons

Chapter 2
Guidelines, Principles and Theories

Guideline: a set of suggestions by an expert

Navigating the interface:
- Standardize task sequences (users may be allowed to perform tasks in the same sequence across similar conditions)
- Ensure that embedded links are descriptive
- Use unique and descriptive headings
- Use check boxes for binary choices
- Develop pages that will print properly
- Use thumbnail images to preview larger images

Organizing the display:
- Consistency of data display (terminology, abbreviations, formats, colors, capitalization, etc.)
- Efficient information assimilation by the user (e.g. left justification for alphanumeric data, right justification for integers, lining up of decimal points)
- Minimal memory load on the user (users should not be required to remember information from one screen for use on another screen)
- Compatibility of data display with data entry
- Flexibility for user control of data display (e.g. users may be allowed to sort using suitable column)

Facilitating data entry:
- Consistency of data-entry transactions
- Minimal input actions by user
- Minimal memory load on users
- Compatibility of data entry with data display
- Flexibility for user control of data entry

Principle: a fundamental law

Determine users' skill levels:
- Novice or first-time users
- Knowledgeable intermittent users
- Expert frequent users

Identify the tasks:
- Frequent actions (use single special keys like arrows, PgUp, PgDn, Ins, Del)
- Less frequent actions (use Alt and Ctrl combinations)
- Infrequent actions (use menu system or form filling)

Choose an interaction style:
- Direct manipulation (objects and actions represented by icons and menu items, respectively) (proposed Ben Shneiderman)
- Menu system
- Form filling

Theory: broad, tested and reliable rule

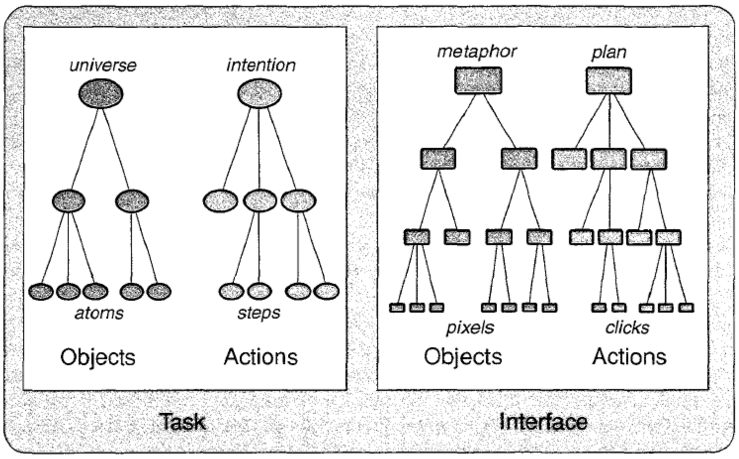
Stage-of-action model (proposed by Norman)
Use the following seven steps iteratively.
1. Forming a goal (e.g. I want to copy a file)
2. Forming an intention (e.g. I have to drag the file from one folder to another)
3. Specifying an action (e.g. sequence of mouse action)
4. Executing an action
5. Perceiving the system state
6. Interpreting the system state
7. Evaluating the outcome

GOMS model (proposed by Card, Moran and Newell)
Goals, operators, methods and selection rules

Keystroke-level model (proposed by Card, Moran and Newell)
Keystrokes, mouse movement and clicks

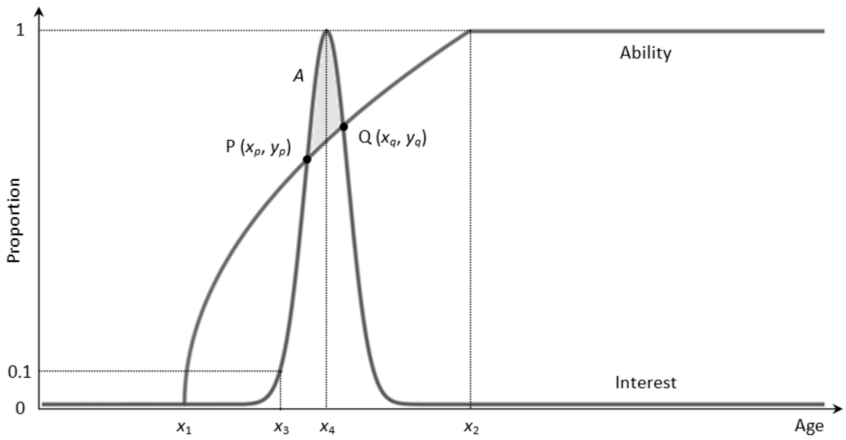
Context-of-use theories

Object-action interface (OAI) model (proposed Ben Shneiderman)



E.g. the universe is a university with multiple departments having students. Actions will be needed at various levels.

Ability and interest model (proposed by Yadav and Chakraborty)



Topic T3
Properties of Good Interaction Design

1. Discoverability
— User should be able to figure out what can be done and how
— User should be able to discover functions rather than learning them from elsewhere

2. Simplicity
— Only relevant information and functions should be displayed

3. Affordance
— User should be able to figure out how to use an object from its properties

4. Mapping
— Use terms familiar to user rather than system-oriented terms (e.g. cut, copy, paste)
— Effect of a function should be clear to user

5. Perceptibility
— System should keep the user informed about what is going on
— User should be able to perceive the current state of the system

6. Consistency
— There should be consistency of design within the system and with other systems
— Lessons learnt by the user while using one system should be applicable in other systems (e.g. hyperlinks are shown in blue, standard names of menu and grouping of menu items)

7. Structure
— Design should organize the interface purposefully
— Group similar functions together and separate dissimilar functions

8. Flexibility
— There may exist multiple ways of performing a task
— Accommodate users with different abilities and purposes

9. Ease
— System can be used efficiently and comfortably with minimal fatigue (e.g. minimize keystrokes, mouse movement and clicks)

10. Error prevention
— Prevent users from making mistakes (e.g. enter new password and then confirm password)

11. Tolerance
— Reduce cost of mistakes (e.g. emergency exits, undo-redo options)
— This allows user to make use of the discoverability property

12. Audience specific
— Interaction should be according to the maturity of user and seriousness of the task

13. Customizable
— User should be able to customize some aspects of the interface (e.g. font, size and color of text)

14. Feedback
- Provide immediate and informative feedback
- Provide feedback on both success and failure
- No codes in error messages
- Poor feedback may be irritating and provoke anxiety

15. Troubleshooting
- User manual should be the last resort
- User manual should not just describe the interface
- User manual user-centric, task focused and easy to search

Users complain about bad HCI. Good HCI is invisible.

Chapter 3
Managing Design Processes

Usability engineering: how to assess usability of products and how to design better usable products?

Write down a guideline for your project:

- Words, icons and graphics
  - Terminology (objects and actions), abbreviations and capitalization
  - Character set, fonts, font sizes and styles (bold, italic and underline)
  - Icons, buttons, graphics and line thickness
  - Use of color, backgrounds, highlighting and blinking

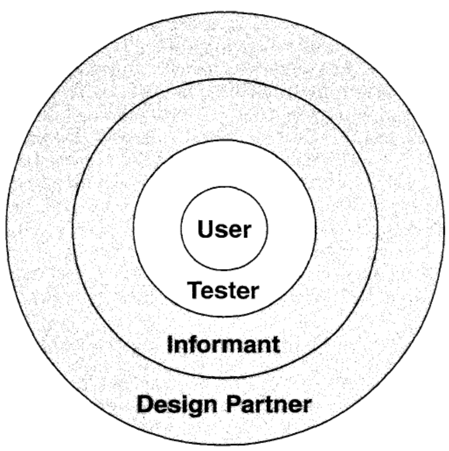
- Screen layout
  - Menu selection, form filling and dialog-box formats
  - Wording of prompts, feedback and error messages
  - Justification, whitespace and margins
  - Data entry and display formats for items and lists
  - Use and contents of headers and footers

- Input and output devices
  - Keyboard, display, cursor control and pointing devices
  - Audible sounds, voice feedback and touch input
  - Alternatives for users with disabilities

- Action sequences
  - Direct-manipulation (clicking, dragging and dropping and gestures)
  - Shortcuts and programmed function keys

Logical User-Centered Interactive Design (LUCID) methodology:
1. Envision
2. Discovery (what users think and require)
3. Design foundation (screen-level prototype)
4. Design detail
5. Build
6. Release (roll-out plan and usability test)

Participatory design

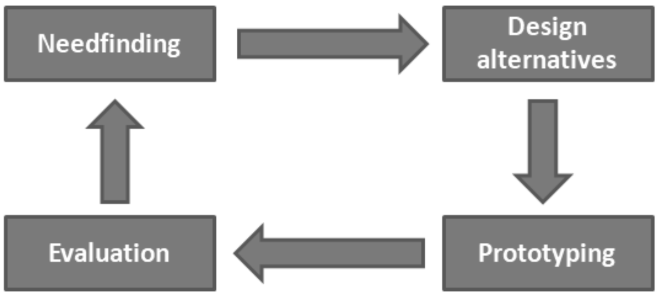


Ethnographic observation: how different types of users use the product?
Scenario development: hypothetical analysis of how a particular user will solve a particular problem

Topic T4
User-Centered Design

Design should not only fulfill the technical requirements but also consider the human factors. User-centered design approach considers the needs of users throughout the development process.

Six principles of user-centered design prescribed by ISO:
1. The design is based upon an explicit understanding of users, tasks and environments
2. Users are involved throughout design and development
3. The design is driven and refined by user-centered evaluation
4. The process is iterative
5. The design addresses the whole user experience
6. The design team includes multidisciplinary skills and perspectives



Data from users: Quantitative and qualitative
Quantitative data: Nominal, ordinal, interval, ratio
Qualitative data: Transcripts, field notes
Qualitative data can be converted quantitative data through some form of coding
HCI uses both quantitative and qualitative data

I. Needfinding

Questions:
- Who are the users and other affected people?
- Where are the users?
- What are their goals and sub-goals?
- In what context they the using the system?

Understand the entire problem space and not only the interface
Avoid confirmation bias
Observations:
- Naturalistic observation: Observe an user doing the task
- Participant observation: Perform the task

Methods
- Interview:
  - Ask open-ended and semi-structured questions
  - Avoid question that have yes/no answers or one-word answers
  - Let the user talk more
  - Interview is not conversation
- Focus groups: Interviewing multiple users together
- Think aloud: User verbally describes how he/she is solving a task

- Post-event protocol: User solves a task and then explains what he/she just did
- Survey: Highly scalable
- Schedule: If you think that the users cannot express themselves properly

II. Design alternatives
- There typically exists multiple ways of solving a task
- Explore multiple alternatives
- Avoid allegiance to existing systems
- Avoid tunnel vision, *viz*. generalizing a particular viewpoint
- Analyze from user's perspective

III. Prototyping
Build a representation of the system, put it in front of users and gather feedback
We need feedback quickly
We build more sophisticated prototypes iteratively

Early and low-fidelity prototypes → Feedback on functionality
Late and high-fidelity prototypes → Feedback on performance

Horizontal prototype → Shallow representation of the entire system
Vertical prototype → Detailed representation of a particular module of the system

Types of prototypes:
- Paper prototype
- Card prototype
- Wireframe prototype
- Working prototype

IV. Evaluation
Evaluation is done first in lab and then in field
Initially, feedback is qualitative and formative
Later, feedback is quantitative

Qualitative feedback
- Based on live demonstration of a prototype (e.g. if a user likes/dislikes a feature, if a user finds it easy/difficult to use a feature)

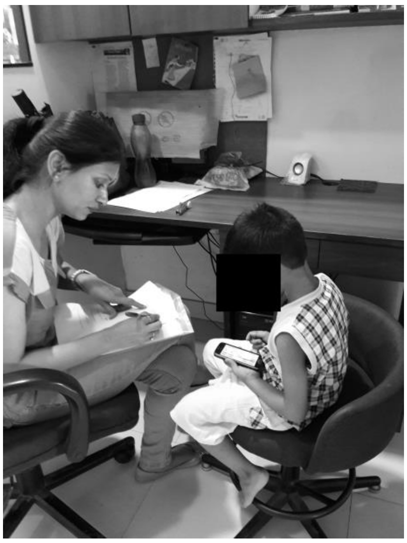
Quantitative feedback:
- What to measure? Accuracy, efficiency, learnability, satisfaction, etc.
- Quantitative data needs to be tested statistically
- Verify that the difference in results for different treatments is repeatable and not random

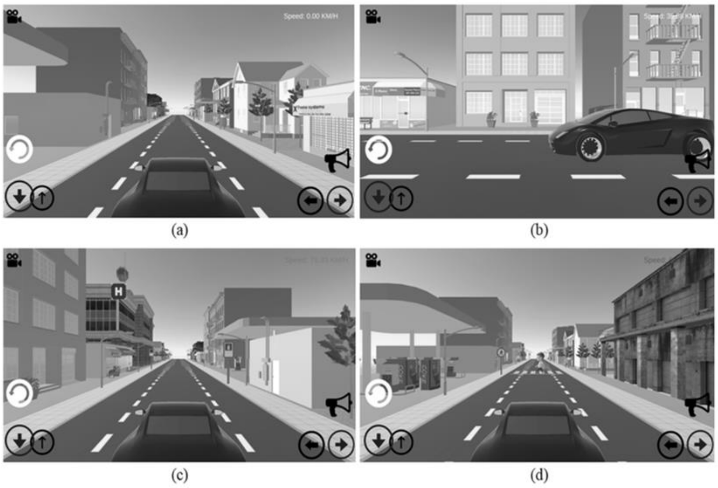
If there are multiple prototypes, then present them to the users in a random order to negate bias

Case study: *Safe Driving* app

We hired 10 children as design partners and 30 children as testers.

Iteration	Need finding	Design alternatives	Prototyping	Evaluation
Iteration 1	Interview 10 children: What type of car racing themed game apps children use? How much time children spend playing with those apps?	We consider several game play alternatives and select one.	Paper prototype: We write down a description of how the app will behave.	We ask the children if they like the concept and what features they would like to have in the app.
Iteration 2	What features children like in car racing themed game apps?	We consider the different features and select a set of features.	Wireframe prototype: We draw dummy screens using Adobe Xd.	We seek feedback on the design of the screens from the children.
Iteration 3	How to improve the look and control of the app?	We consider different screen layouts and widgets.	Working prototype: We develop a fully operational app.	We seek feedback on the game play from the children. We also collect empirical data on how the children are using the app.
Iteration 4	How to improve satisfaction among users?	We optimize the game play, screen layout and widgets.	Final app: We optimize the user interface of the app.	We also collect empirical data from 30 children on how they are using the app.





Screenshots of Safe Driving App showing (a) the starting configuration, (b) a side view of the car, (c) a hospital in front of which honking is prohibited and (d) a pedestrian crossing the road who may not be hit.

Chapter 4
Evaluating Interface Designs

Designers can become so entranced with their creations that they may fail to evaluate them adequately.
Experienced designers have attained the wisdom and humility to know that extensive testing is a necessity.

Technical factors that influence the evaluation plan:
- Stage of design (early, middle or late)
- Novelty of project (well-defined versus exploratory)
- Number of expected users
- Criticality of the interface

Expert reviews (at any stage)
- Heuristic evaluation
- Guideline review
- Consistency inspection
- Cognitive walkthrough
- Formal usability inspection

Usability testing
Usability laboratories (are they still relevant?)
Informed consent and Helsinki Declaration
Video-recording
Metrics and statistical analysis of data

User surveys
Collect demographic information of the users and their views on the user interface
Likert scales

Acceptance testing
- Training time
- Speed of task performance
- Rate of errors by users
- User retention of commands over time
- User satisfaction

Observation schedules
Interviews and focus group discussion
Data logging
Suggestion box and error reporting
Discussion groups and newsgroups
Controlled psychologically oriented experiments

Chapter 5
Software Tools

Specification methods for user interfaces: natural language, semi-formal, formal, transition diagrams

Menu selection trees
Dialog box trees

A branch in a transition diagram may be labeled with its probability

Interface building tools separate user interface design and backend programming.

Interface building tools:
- Hand-drawn
- Drawn using MS PowerPoint
- Designed using IDE
- Specialized software

Evaluation tools
Probes

Chapter 6
Direct Manipulation and Virtual Environments

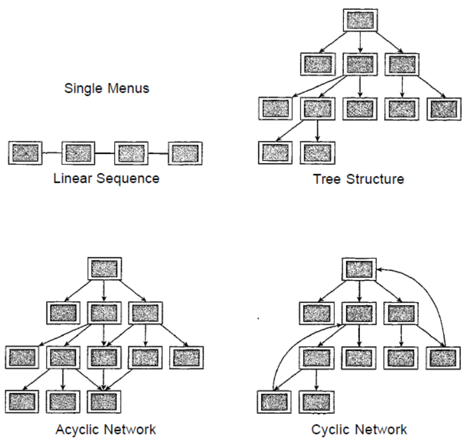
The direct manipulation approach was developed as an alternative to command language approach.
Objects and actions are represented by icons and menu items, respectively.
Continuous representation of objects together with rapid, reversible and incremental actions, and feedback are used.
The interface is more natural or intuitive.
By the early 1990s, the “what you see is what you get” or WYSIWYG paradigm became common.
The desktop metaphor was proposed by Ben Shneiderman.
GUI comprises of windows, icons, menus and pointer (WIMP).
Visual descriptors are preferred over text.
Different types of widgets are used.
Feed forward is also used at places.

User can:
- Select an object
- Modify its representation
- See its properties
- Perform actions on it
- Work with multiple objects

Touch-based interaction
Three-dimensional user interface
Augmented reality
Virtual reality

Chapter 7
Menu Selection, Form Filling and Dialog Boxes

When designers cannot create appropriate direct-manipulation strategies, menu selection and form filling are attractive alternatives.
Menu systems require recognition rather than recall.
Form filling requires recall.



In a hierarchical menu system, time required to select an item = k + c * log b, where b is the number of branches at each level.

Pull-down menu
Pop-up menu (context-aware, less mouse movement)

Factors to decide the ordering of items in a menu:
Alphabetical or numeric ordering
Logical grouping with line between groups
Most frequently used items first
Most important items first

Adaptive menu systems (initially display only the frequently used items and display remaining items if necessary)
Useful if here are a large number of items

Keyboard shortcuts may be defined for menu items

Form-filling design guideline:
- Meaningful title of form
- Comprehensible instructions
- Logical grouping and sequencing of fields
- Familiar field labels
- Consistent terminology and abbreviations
- Visible space and boundaries for data-entry fields
- Convenient cursor movement
- Error messages for unacceptable values
- Marking of compulsory/optional fields

Many tasks are interrupted to request users to select options or perform limited data entry.

The most common solution is to provide a dialog box.
Familiar examples include the Open, Save, Find, Print and Font.

Dialog box design guideline:

- Issues similar to those of form filling
- Distinguishable from the background
- Standard buttons (OK, Cancel)
- Should not hide relevant content
- Possible to make disappear

Audio menu
Menus for small displays (mode/advanced/triple-dots button)

Chapter 8
Command and Natural Languages

Command language
— Command and argument format

Natural language interaction
— Natural language queries
— Audio response

Chapter 9
Interaction Devices

Keyboards and keypads
- Layout
- Auto-repeat feature of keys
- Printable character keys
- Enter, Tab
- Shift, Alt, Ctrl
- Caps lock, Num lock, Scroll lock (when the Scroll Lock is on, then the arrow keys scroll the contents of a text window instead of moving the cursor)
- Arrow keys, Home, End, PgUp, PgDn, Ins, Del
- Function keys (fixed or programmed tasks)
- Widows key, menu key
- PrtScr, Pause

Pointing devices:
Direct control devices (light pen, stylus, touchscreen)
Indirect control devices (mouse, joystick, touchpad)

Fitts' law
$ID = \log_2 (2D/W)$
$MT = a + b\ ID$

Displays (including large and small displays)
Printers (dot matrix, inkjet, laser jet)
Plotter
3D printer
Braille embossers

Chapter 10
Collaboration

Synchronous collaboration
- Text-based chat
- Video calls, online meeting

Asynchronous collaboration
- Wiki
- Social media
- Online office suites
- Online compilers

Topic T5
Graphical User Interface

Icons:
- Entities (file, folder, etc.)
- Actions (cut, copy, paste, print, etc.)
- Abstract symbols (equation icon of MS Word, insert sound icon of MS PowerPoint)

Navigation:
- Logically organize the sequence of actions across screens
- Possible to revisit a previous screen
- Widgets: buttons, scrollbar, tab

Data entry:
- Provide an easy way to enter data
- Similar sequence of actions to enter data throughout the system
- Minimize keystrokes, mouse movement and clicks
- Tab key for moving focus among widgets
- Widgets: textbox, radio button, checkbox, slider, list box, combo box

Information display:
- Facilitates assimilation by intended users
- Mix of text and illustrations
- Carefully selected fonts, sizes and colors
- Same style throughout the system
- Use format that was used at the time of data entry
- Display all relevant information in a screen so that the user need not remember what was displayed in the previous screen
- Widgets: label, tool tip, status bar, progress bar and widgets used for data entry in disabled mode

Orientation:
- Aspect ratio = width : height
- Portrait, landscape

Pixel = picture element
- Square shaped
- Color depth = bits per pixel

Color:
- Text, highlight, shape border and fill
- 24-bit RGB model
- Transparency

Bold, italic, underline
Strikethrough
Superscript, subscript
Case: upper case, lower case, capitalize each word, small caps
Alignment: left, right, center, justify
Line spacing: single, double, custom

Word wrap
Text wrapping of figures

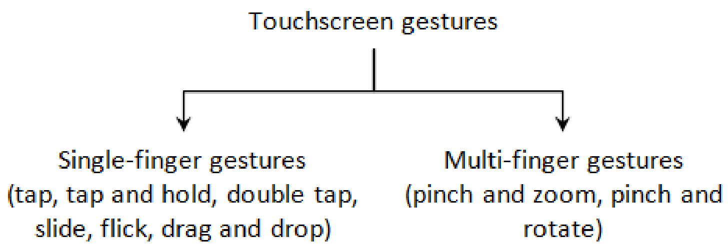
Typeface:
- A set of glyphs (purposeful marks) each representing a character
- Serif and sans serif typefaces
- Monospaced (fixed-width) and proportional typefaces
- A font is a digital representation (file) of a typeface
- Bitmap fonts and vector fonts (also known as outline fonts)
- Vector fonts can be displayed and printed in any size
- TrueType is a vector font standard
- Times Roman and Times New Roman fonts represent the same typeface; same is true for Helvetica and Arial and Courier and Courier New

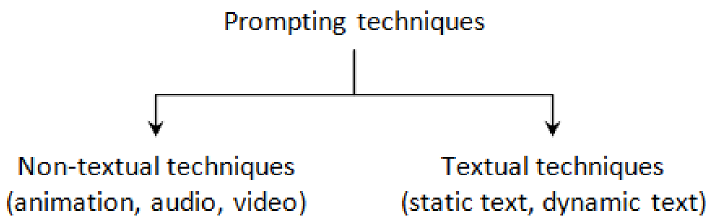
Font size:
- Four line notebook: capital line, waist line, base line, long line
- Font size = x-height = distance between waist line and base line
- 1 point = 1/72 inch

Kerning:
- Negative kerning (e.g. AV)
- Positive kerning (e.g. CO)

Ligature:
- Th, fi, et

Text figures:
- To make numbers look like text
- 0123456789





GUIs played a key role in popularizing computers.

Chapter 11
Quality of Service

Response time:

- <1 second for frequent tasks
- 2-4 seconds for common tasks
- 8-12 seconds for complex tasks
- >15 seconds is frustrating
- Variance should be low
- Inform users about long delays

User planning time (time to plan an action)
User think time (time to understand the response)

Shorter response time leads to shorter think time
Faster interaction increases productivity and error rate too

Chapter 12
Balancing Function and Fashion

Error messages:
- Specific
- Constructive
- User-centered phrasing

Multiple window display:
- Dependent windows

Chapter 13
User Manuals, Online Help and Tutorials

Degree of integration in the interface:
- Online manual and tutorial: independent interface, even possibly developed by a different company
- Online help: integrated into the interface, separate window usually invoked from a “help” button
- Context-sensitive help: (a) user-controlled-depends on where the user points (pop-up box, screen tip text) or (b) system initiated (the system makes suggestions)
- Animated demonstrations and guides: usually integrated into the interface

Time of intervention:
- Before starting (manual and tutorial)
- At the beginning of the interaction (getting started, animated demonstration)
- During the task (context sensitive, either user- or system-initiated help)
- After failure (help button, FAQs)
- When the user returns the next time (startup tips)

Online and paper versions

A glossary of technical terms may be helpful.
Appendices with error messages may be helpful.

Chapter 14
Information Search and Visualization

Degree of integration in the interface:
- Natural language queries
- Voice queries
- Form filling queries

Five steps of searching:
1. Formulation
2. Initiate action
3. Review results
4. Refinement
5. Use

Topic T6
Child-Smartphone Interaction

1. Fitts' law is not valid for children's interaction with touchscreen. Onscreen movement time is higher for drag and drop gesture than for tap gesture.

2. Children begin performing single-finger touchscreen gestures between four and six years of age and multi-finger touchscreen gestures between seven and eight years of age.

3. Children begin following non-textual prompting techniques between four and six years of age and textual prompting techniques between seven and eight years of age.

4. Children aged seven to ten years take almost equal time to read from computer, smartphone and printed sheets.

5. Children are able to recall what they have read from computer, smartphone and printed sheets with equal ease.

6. Children aged nine and ten years use hyperlinks more often when reading from smartphone than when reading from computer.

7. Children can read text written in serif typeface and sans-serif typeface on smartphone with equal ease.

8. Children develop ability to use augment reality software by seven years of age.

9. Children aged four to eight years have longer conversations on video calls than on audio calls.

10. Children use bodily gestures more often on video calls than on audio calls.

11. Children pause the person they are conversing with to make a point of their own more frequently on video calls than on audio calls.

12. Children can initiate audio calls and video calls with equal ease.

13. Children find it easier to recall things they have heard on video calls than on audio calls and the difference is more prominent for younger children.

14. Children aged seven and eight can solve real world problems when receiving instructions on a video call.

15. Children aged four to eight years prefer watching videos of toys and games and cartoons, while children aged nine and ten years also like to watch do-it-yourself and educational videos.

16. Children aged between four and eight years prefer watching shows on TV, while children aged nine and ten years like watching videos on smartphones.

17. Children aged between four and ten years prefer voice search over textual query based search, but only children aged seven years and more can use it effectively to search for videos on YouTube.

18. Children learn to use features like the play/pause button and the seek bar, change the volume and skip through advertisement while watching YouTube videos by seven years of age.

19. The average attention span of children aged four to six years, seven and eight years and nine and ten years while watching a video on a smartphone is approximately 3 minutes, 4 minutes and 9 minutes, respectively.

20. Children aged two to twelve years can draw the same subjects with crayons on drawing sheets and with smartphone apps, but their drawings are less detailed when they use apps.

21. Children learn to use various features of drawing apps between two and twelve years of age and want only those features that they can use in the apps.

22. Children aged four to ten years preferred voice search over textual query based search.

23. Vocabulary apps can be used to teach novel words and their pronunciation to children aged four to six years.

24. Children aged eleven and twelve years can be attracted to mathematics lessons using a smartphone app but doing so improves their performance only marginally.

25. A majority of children aged four to ten years use smartphones for more than an hour per day.

26. Overusing smartphone makes children susceptible to throwing tantrums and restlessness, but does not affect their emotional stability and ability to interact with peers.

27. Children aged four to ten years are often exposed to violent text and graphics when using smartphone.

28. Children can serve as design partners for developing smartphone apps for children of the same age.

29. Smartphone apps developed following user centered design are appreciated by children.

30. Children aged four to six years typically click on the first option when challenged with a multiple choice question.

***