Evolution of Computing

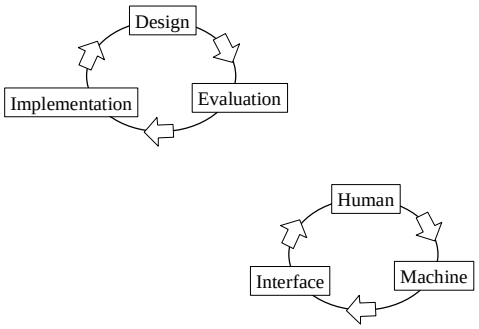
- 1950's
  - computers first appeared onto the commercial scene
  - difficult to use, cumbersome, unpredictable
- 1970's
  - first personal computers
  - provided interactive computing power for individual users at a low cost
  - wide variety of people began using computer systems
  - Man-Machine Interface (MMI)
- Mid 1980's
  - Human-Computer Interaction (HCI)

Definition of HCI

Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them

Multidisciplinary Science

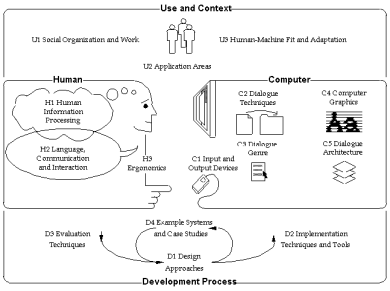
- Computer science
  - provide knowledge about the capability of the technology and idea about how this potential can be harnessed
- Psychology and cognitive science
  - understanding human behaviour and the mental processes that underlie it (information processing)
  - understanding the nature and causes of human behaviour in the social context
- Sociology and anthropology
  - interactions between technology, work, and organization
- Industrial design
  - interactive products
- Ergonomics
  - understand the user's physical capabilities
- Linguistics; artificial intelligence; business; graphics design; technical writing



Computer Science Perspective

- Focus on the ***interaction***
  - between one or more *humans*
  - and one or more computational *machines*

Taxonomy of HCI



Use and Context of Computers

Problems of fitting computers, their users, and the context use together

- Social organization and work
  - understand the nature of work and how human and technical systems mutually adapt
    - humans are social beings
    - models of human activity (small groups, organizations)
- Application areas
  - characteristics of application domains
    - document processing (word processors, spreadsheets)
    - communication tools (email, conferencing, news)
    - embedded systems (TV's, VCR's, microwave)
- Human-machine fit and adaptation
  - improve the fit between the designed object and its use
    - systems adapt to users (customization)
    - users adapt to the systems (training, ease of learning)
    - user guidance (help, documentation, error-handling)

Human Characteristics

To understand how humans process information, structure actions, communicate, and physical and psychological requirements

- Human information processing
  - theory of memory, perception, motor skills, attention, vigilance, problem solving, learning and skill acquisition, motivation
  - human diversity
- Language, communication and interaction
  - syntax, semantics, pragmatics, conversational interaction, specialized languages
- Ergonomics
  - physiological characteristics of people and their relationship to workplace and the environment
    - arrangement of displays and controls; cognitive and sensory limits; effects of display technologies; fatigue and health; furniture and lighting; design for stressful and hazardous environments; design for the disabled, ...

Computer system and interface architecture

Specialized computer components for interacting with people

- Input and output devices
  - mechanics of particular hardware devices and performance characteristics
- Dialogue techniques
  - basic software architecture and techniques for interacting with humans
    - dialogue inputs; outputs; interaction styles
- Dialogue genre
  - conceptual uses to which the technical means are put
    - interaction and content metaphors; transition management; aesthetics

Computer system and interface architecture

Specialized computer components for interacting with people

- Computer graphics
  - computer graphic concepts useful to HCI
    - 2D and 3D geometry; modeling; colour representation
- Dialogue architecture
  - software architecture and standards
    - window managers; interface toolkits; look and feel; standardization

Development Process

The construction and evaluation of human interfaces

- Design approaches
  - graphic design basics; software engineering; task analysis; industrial design
- Implementation techniques and tools
  - tools for implementing interfaces
    - prototyping techniques; object-oriented methods; application/device independence
- Evaluation techniques
  - philosophy and scientific methods of evaluation
    - productivity; measure of merit (ie; time, errors, learnability, preference); formative and summative evaluation techniques (ie. field studies, interviewing, questionnaires, system logging)
- Example systems and case studies

HCI Goals

- To produce
  - usable systems
  - safe systems
  - functional systems
- Usable vs. Useful

Why are user interfaces important in today's computer systems?

Why are user interfaces important?

- Business view:
  - help human be more productive/effective
  - human costs outweigh hardware and software costs
- Personal view:
  - people want computers to perform as appliances
- Marketplace view:
  - wide variety of people now using computers
    - expect "easy to use" systems
    - not tolerant of poorly designed systems
    - heterogeneous group
  - if people aren't satisfied they will chose a different product
- System view:
  - complexity of humans, computers, and interface between the two

Why are user interfaces important?

- Human factors view:
  - humans have limitations and errors are costly
- Social view:
  - Computers contribute to critical parts of our society and cannot be ignored
    - educate our children
    - take medical history and provide expert advice
    - handle credit and financial information
    - control air traffic control and ground traffic flow
    - book travel
    - control chemical/oil/nuclear plants
    - control space missions
    - assist humans with everyday tasks (office automation)
    - control complex machines (aircraft, space shuttles, super tankers)
    - help control consumer equipment (cars, washing machines, microwave)
    - entertainment

Motivations

- Life-critical systems
  - Air traffic control, nuclear reactors, power utilities, police & fire dispatch systems
- Industrial and commercial uses
  - Banking, insurance,order entry, inventory management, reservation, billing, and point-of-sales systems
- Office, home, and entertainment applications
  - Word processing, electronic mail, computer conferencing, and video game systems
- Exploratory, creative, and cooperative systems
  - Database, artist toolkits, statistical packages, and scientific modeling systems

Why are user interfaces difficult to design?

HCI Challenges

- How to keep abreast of changes in technology
- How to ensure that their designs offer good HCI as well as harnessing the potential functionality of the new technology.

Why have an interface design process?

63% of large software projects go over cost

- managers gave four usability-related reasons
  - users requested changes
  - overlooked tasks
  - users did not understand their own requirements
  - insufficient user-development communication and understanding
- Usability engineering is software engineering
  - pay a little now or a lot later
  - too easy to jump into a detailed design that is:
    - founded on incorrect requirements
    - had inappropriate dialogue flow
    - not easilty used
    - never tested until it is too late

What are the most common user interface techniques uses in today's systems?

Direct manipulation of graphical objects

- Sketchpad (Sutherland, 1963)
  - manipulation of objects using a light pen
  - grabbing objects, moving them, changing size and using constraints
- Reaction Handler (Newman, 1966/1967)
  - first "widget"
- AMBGIT (MIT Lincoln Labs, 1968)
  - iconic representation, gesture recognition, dynamic menus, selection of items by a pointing device
- Pygmalion (Canfield, 1975)
  - coined the term "icons"
- Bravo & Draw (Xerox PARC, 1970's)
  - how objects and text are selected, opened and manipulated
  - WYSIWYG
- Dynabook (Kay, 1977)
  - direct manipulation interfaces for everyone
  - followed by commercial systems: Xerox Star, Apple Lisa, and the Macintosh
- Shneiderman (1982)
  - coined the term "direct manipulation"

Mouse, Windows, Hypertext and CSCW

- Englebart (1968) - Augmenting Human Intellect
  - Mouse
  - Multiple tiled windows
  - hypertext
  - CSCW - remote participants at various sites
- Englebart (1962)
  - Word Processor
    - automatic word wrap, search and replace, macros, scrolling text, move, copy and delete characters, words or blocks of text
- Alan Kay
  - Overlapping windows (1969)

Early Applications

- Drawing
  - Sketchpad (Sutherland)
  - Superpaint, MacPaint, and MacDraw
- Text editing
  - Englebart (1962)
  - Bravo (WYSIWYG)
  - Star, LisaWrite, MacWrite
- Spreadsheets
  - VisiCalc (Frankston and Bricklin 1977-1978 MIT)
- Hypertext
  - Vannevar Bush - MEMEX
  - Ted Nelson coined the term (1965)
  - Englebart (1968)

Two most influential commercial graphical user interface systems:

- XEROX STAR (1981)
- APPLE LISA (1982)

The best interface design work in the world may be wasted if it is not tightly coupled with a timely product that offers significant functionality at a reasonable price, skillfully marketed to appropriate customers.

8010 Star Information System

- conceived in 1975 and released in 1981
- "new personal computer designed for offices … intended for business professionals who handle information
- System Development Division at Xerox
- Alto - Xerox Palo Alto Research Center (PARC)
- bitmapped screen, windows, mouse-driven interface, icons
- changed the notion of how interactive systems should be designed
- high quality interface
- many of the design aspects were done right

Xerox Star - user interface features

- desktop metaphor
  - users conceptual model
- direct manipulation
  - emphasizing recognition over recall
- property or option sheet
  - to specify the appearance of objects
- WYSIWYG (what you see is what you get)
- generic commands with dedicated keys
- high degree of consistency
- few modes
- icons and iconic file management
- progressive disclosure

Emphasis on good graphic and screen design

- Appearance and placement of screen objects
- illusion of manipulable objects
  - usual task is to present information for passive viewing
  - needed to present information for manipulation as well
- visual order and user focus
  - intensity and contrast to draw the user's attention to the most important features of the display
- revealed structure
  - can show "structure" or "non-printing characters"
- consistent and appropriate graphic vocabulary
- match the medium
  - work within the constraints of the system

Xerox Star - features

- Distributed computing
  - connect personal workstations with a local area network and attach shared resources to the network
- mouse
  - Star handled the mouse at a very low level to improve performance
  - two button mouse
- bitmapped display
- windows
  - first commercial system to provide windows
  - issue of overlapping windows
- integrated applications
  - focus was on document processing so all tools were integrated into that environment

Xerox Star - Why wasn't it successful?

- Market not available? (trailblazer?)
- cost?
  - $15,000
  - star's benefits were not perceived to be worth the additional cost
- limited functionality
- lacked an open architecture
- perceived as slow

The Apple Lisa

- Conceived in 1978, released in 1983
- a product with a similar interface to the Star
  - it is said that Steven Jobs, Apple's founder and chairman, visited Xerox PARC in 1979
- the Star was more ambitious than the Lisa in networking and distributed computing
- Lisa was positioned between an office system and a personal productivity tool
- less expensive ($10,000)

Lisa - user interface features

- desktop metaphor
  - typical user was a business person whose day was constantly interrupted with immediate requests to do other things
- mouse & windows (after Xerox visits)
- consistency between all Lisa applications
- multi-tasked environments
  - multiple windows to display different types of work
- users select document and Lisa would determine the application needed

Lisa - Desktop managers

- Desktop icons
  - display too small
  - dragging would be cumbersome if can't find wastebasket
  - locating documents in nested folders difficult
- Document browser
  - rejected hierarchical filing because wanted to ease placing/finding a document
  - used attribute filters to help display appropriate documents
    - was not obvious and was difficult for some operations
- Twenty Questions Filer
  - dialog prompt for document parameters
    - system faster and more accurate but was still abstract and wasn't FUN!
- Dataland
    - too difficult to manage large number of documents
- Desktop icons (direct manipulation)
  - simplicity and approachability!

Lisa - critical factors

- from the beginning - focus on the USER!
- interface developed through experience, not programmer intuition
- extensive user testing on representative users

- withdrawn from the market after three years
  - cost
  - confused product positioning
  - inadequate application base

The Apple Macintosh

- January 1984
- approximately $2,500
- SUCCESSFUL!
  - Did not need to trailblaze
  - a second-general Lisa and could learn from previous experience
  - aggressively priced
  - partially open architecture and a powerful developer's toolkit
    - lead to widespread availability of software applications
  - desktop publishing market
    - excellent graphics and reasonably priced laser printers
  - marketing experience, distribution channels, and experience in sales and support

Now PC's are dominating the computer market over Macintosh. Why?

A Case Study in Interface Design

The CHI '89 Information Kiosk
Gitta Salomon

Project Goals

- to explore the interface design process
- make use of multimedia
- expand the team's understanding of what constitutes a successful interface design
  - through observation of a wide range of people interacting with the system

An overview of the system

** Insert Figure 1, page 25 **

Design Methodology

- Iterative design process, based on successively enhanced prototypes
- initial design specification phase
  - rough paper sketches and screen mock-ups were created
- storytelling prototype phase
  - the earlier designs were refined and used to "tell stories" about the system's functionality to others
- functional prototype phase
  - the prototypes were make semi-functional and informal testing was done
  - uncovered several problems which significantly effected usability

Initial design specification phase

- early designs were based on the team's perspective
- Used visual specifications as opposed to textual
  - early design were communicated with through paper sketches or HyperCard mock-ups
  - often made use of visual placeholders to demonstrate possible screen design, functionality and sequencing
  - roughness of the design prompted more modifications
- Design sessions around the computer
  - easy modifications
  - use visual methods to demonstrate and idea
  - annotate problems directly on the interface
  - maintained the perspective of the user

Storytelling prototype phase

- Prototypes from the initial phase were expanded on to create sets of screens showing interaction sequences
- The enhanced designs were then show to colleagues and friends (outside of the project)
  - presenter would describe the process and events
  - could also elicit feedback by asking what would happen next

Functional prototype phase

- semi-functional prototypes were informally tested with users and quickly reworked based on feedback
  - goal was to improve the interface as much as possible through rapid redesign and retesting, before all of the functionality was put in place
- discount usability testing
  - subjects asked to perform specific tasks and items of interest
  - and the end of each session an interview was used to collect general impressions
- most of the problems uncovered were related to misinterpreted functionality and were relatively easy to correct through visual representation

Example 1: representing functionality

** Insert Figure 4(e), page 28 **

Example 1: Solution

** Insert Figure 4(f), page 28 **

Example 2: The “Summary” button

** Insert Figure 4(e), page 28 **

Example 2: Solution

** Insert Figure 4(f), page 28 **

Example 3: Yearbook navigation

** Insert Figure 7, page 31 **

Example 3: Solution …. or not?

** Insert Figure 8, page 31 **

Use of trace data

- To explore usage patterns to uncover where, in practice, the interface design was successful and where it was flawed
  - unobtrusive
  - anonymous
  - difficult to capture when one session ended and the next began
  - over 5,600 files were created
- Design of selectable items:
  - 21% of users clicked on the field containing a bulleted list (which wasn't active!!)
  - similar behaviour found on other cards
  - design team had carefully established conventions about clickable items ….
  - Indicates that bulleted lists might carry expectations of interactivity in the computer domain

Use of trace data (continued)

- Intelligent defaults
  - designers assumed that when performing a time sort, the user would be most interested in the current or near future time slots
  - verified by the trace data
    - in more than half the cases, users made their selection from the screen they were shown first
- Showed the lack of use of the index button (only 3%)
- Improving trace data
  - capture coordinates of a mouse click for more precise information of what was clicked on
  - these types of additions also add to the difficulty of analyzing the trace data

Principles used in this case study:

- user-center design
- progressively more refined designs and prototypes
  - benefits of low-level, medium-level and high-level prototypes
- early and frequent user testing
- three design stages
  - initial design specification
  - storytelling prototype phase
  - functional prototype phase
- system logging

Readings for Tuesday, Jan. 30th

- Getting to know users and their tasks
  - Clayton Lewis and John Rieman
  - on-line document

  ftp://ftp.cs.colorado.edu/pub/cs/distribs/clewis/HCI-Design-Book/chap-2.v-1
- How to design usable systems (excerpt)
- Learning from Notes: Organizational Issues in Groupware Implementation