Course Overview

- **Introduction**
- Understanding Users and Their Tasks
- Principles and Guidelines
- Interacting with Devices
- Interaction Styles
- UI Design Elements
- Visual Design Guidelines
- UI Development Tools
- Iterative Design and Usability Testing
- Project Presentations & Selected Topics
- Case Studies
- Recent Developments in HCID
- Conclusions

Chapter Overview Introduction

- Logistics
- Motivation
- Objectives
- Relevance of HCID
- Technologies and Ideas
- Evaluation Methods
- Rapid Prototyping
- User-Centered Design
- Important Concepts and Terms
- Chapter Summary

 [Dix, Preece, Mustillo]

Logistics

- Introductions
- Course Materials
- Term Project
- Homework
- Exams
- Grading

Introductions

- Instructor
- TA ??

Instructor

- Dr. Franz Kurfess
- Associate Professor, CS Dept.
- Areas of Interest
  - Knowlege Mangement
  - SoftwareEngineering
  - Neural Networks & Structured Knowledge
- Contact
  - preferrably via email: franz@cs.concordia.ca
  - phone (514) 848 3038
  - office LB 903-21

Humans & Machines

- Briefly write down two experiences with human-machine interface design issues
  - positive
    - intuitive design, efficiency, curiosity, aesthetics, ...
  - negative
    - confusing, techno overload, impractical, counter-intuitive, inefficient, ...

Class Participants

- Name, occupation, interest, background, ...
- Why this course?

Course Material

- on the web
  - syllabus
  - schedule
  - project information
  - project documentation by students
  - homework assignments
  - grades
- address
  `http://www.cs.concordia.ca/~faculty/franz`

Term Project

- development of a practical application in a team
  - prototype, emphasis on user interface, user-centric design
- implementation must be accessible to other class participants
  - e.g. Java Swing, JavaScript, Visual Basic
- three deliverables, one final presentation
- peer evaluation
  - teams evaluate the system of another team
- information exchange on the Web
  - course Web site
  - documentation of individual teams
    - team accounts

Project Theme:

Ubiquituous Computing and Knowledge Management

- "hidden" computers in everyday devices
- utilization of domain and background knowledge
- adaptive user interfaces
  - task-centered
  - user-centered

Homework

- individual assignments
- approximately ten for the whole course
- usually one to two pages of questions, exercises, outlines
- usually no programming

Exams

- one midterm exam
  - in-class
- one final exam
  - take-home

Human-Computer Interaction

- Addresses any interaction by humans with computer systems:
  - as users
  - as developers
  - as individuals
  - as groups
- Also referred to as *User Interface Design*, or *Human-Computer Interface Design*

Human-Computer Interaction (cont)

- Concerned with the **process** of design (i.e., not only the **what**, but also the **how** & the **why** of interface design)
- Is part of the larger discipline of Human Factors or Human Factors Engineering (known as Ergonomics in Europe), which looks at how users:
  - **perform** activities, tasks, & jobs
  - **interact** with systems
  - **use** tools, machines, computers, & software

Relevance of HCID

- the goal is to develop and improve systems so that users can carry out their tasks:
  - **effectively**
  - **efficiently**
  - **enjoyably**
  - **safely** (especially in safety-critical systems like air traffic control)
- These aspects are known collectively as ***Usability**** *(Much more about this later)*

Examples

- Example 1: **Three-Mile Island Nuclear Disaster**
  - Improperly located displays/controls obscured key components from one another.
  - Conflicting feedback messages prevented technicians from understanding what was happening and reacting quickly in the appropriate manner.
- Example 2: **Software Development**
  - In 1995, only 10% -20% of software projects done in the U.S. were successful. This represents a loss of about $80 billion U.S. annually (Standish Group).
  - The IRS recently announced that it had wasted about $4 billion in software development and would have to start again from scratch (Standish Group).
  - The UI accounts for about 50% of current application effort (Myers & Rosson, 1992).

Motivation

- Why are you in this class?
- What do you expect to get out of this class?

Classroom Feedback

Objectives

- Write down three to five goals you want to obtain through this class. Indicate how they relate to HCID.
  - knowledge
  - skills
  - methods & tools
  - experience

Classroom Feedback

Approaches to HCID

- ***Systems engineering approach***
  - engineering model, bottom-up, reductionistic
- ***User-centric approach***
  - (user-task model, top-down, holistic)
  - Example: water faucet design
    - conventional faucet design, separate hot/cold taps
    - single-handle faucet design, integrated flow rate and temperature control mechanism
  - Example: programmable remote control device
    - hexadecimal representation
      - 16 pages of storage, 16 command sequences, 0-F
    - task oriented, common task terms
      - VOLUME, CHANNEL, POWER, STOP

Remote Control Overflow

- usage studies suggest that most users utilize only a small part of the functions
  - typically around 5-7
- technically it is feasible to build controls that can communicate with almost all remotely controllable devices
- try to identify the obstacles towards a universal remote control
  - emphasis on user interface

Classroom Feedback

Scope of HCID

- primary goal is to design ***usable*** systems
- requires knowledge about:
  - **Who** will use the system - the **user**
    - motivation, satisfaction, experience level, etc.
  - **What** will it be used for - the **tasks**
    - office, information retrieval, transaction-based, etc.
  - Work **context** and environment in which it will be used
    - job content, power and influence, personnel policies, etc.
  - What is technically and logistically **feasible**
    - technological capabilities, memory size, costs, time scales, budgets, etc.

HCID Principles

- **Understand** the User and the Application
- **Ensure** Self-Evident Feature Operation
- **Use** Users' Knowledge Across Systems
- Don't Slow Down the User
  - **work** with the user, not against him/her
- Provide **Simple** Ways to Deal with User Errors

User Errors

- Tell the user what the current state of the system is
- Provide a means for the user to get human assistance
- Provide a means to allow the user to back up a step
- Provide a way for the user to return to the beginning of the application

What is a "User Interface"

- The term "user interface" refers to the methods and devices that are used to make thei nteraction between machines and the humans who use them (users) possible
- UIs can take many forms, but always accomplish two fundamental tasks:
  - Communicating information from the machine to the user
  - Communicating information from the user to the machine

What is a "Good" User Interface?

- Ideally, the UI should represent the capabilities of the entire system
- The more complex the system, the more important is the UI
- The UI should help the user build a "mental model", or an intuitive sense, of how the system works
- When a system feels natural to use, the UI is doing a good job
- A good UI helps tailor the system to the user (adaptive)
- A good UI helps users absorb information
- A good UI meets the principal design goals (e.g., learnability vs. usability, first-time use, infrequent use, or expert use, etc.)

Seven Deadly Sins of User Interface Design

1. Design for technology rather than the user
   -> *technology is not the panacea*
2. "Coolness"
   -> *flashy graphics do not improve a bad UI*
3. Logical vs. visual thinking
   -> *users don't think like software designers*
4. User input as right or wrong
   -> *design for error*
5. Overextend basics
   -> *make simple things simple, complex things possible*
6. Fix it with documentation
   -> *users don't read documentation; don't try to fix a UI defect through documentation*
7. Fix it in the next release
   -> *old habits are hard to break*

(Adapted from Trower, 1994)

Confession Time ...

- Which of the seven deadly sins have you committed?
- What were the
  - causes
  - consequences
  - repair (attempts)
- Write down two examples
- Sharing in class is optional

Classroom Feedback

History of HCID

- arranged roughly into decades
- user interface design and related issues
- experimental and commercial systems

Your First Computer Memories

- What was the user interface of your first computer?
- In retrospect, what did you
  - like about it
  - dislike about it
- Jot down the year you started using a computer, and what type it was.
- Compare the year with your neighbour; the one with the older year wins!

1940s

- Increasing complexity of aircraft fighter cockpits and increasing no. of "pilot error" accidents during W.W.II
- Coining of the term "man-machine interface"
- Introduction of the first modern electronic computers
- ENIAC (1943), the world's first all electronic numerical integrator and computer
- Mark 1 (1944), the world's first paper tape reader
- Publication of Vannevar Bush's "As we may think" (1945) article in Atlantic Monthly

1950s

- Introduction of assembly language

1960s

- Invention of the mouse (1963) by Douglas Engelbart at Stanford University
- Ivan Sutherland's SketchPad (1963 Ph.D. Thesis)
  - Introduced many new ideas/concepts now found in today's interfaces (e.g., hierarchical menus, icons, copying, input techniques (light pen), etc.)
- Data tablet (1964) as an input device
- Multiple tiled windows, Engelbart 1968
- Idea of overlapping windows proposed by Alan Kay in 1969 Ph.D. dissertation
- Dynabook (1969) by Alan Kay - the first prototype of a notebook computer

1970s

- Emergence of the first personal computers
  - Altair, Apple
- Start of migration to the desktop

1980s

- First Graphical User Interface (GUI) developed at Xerox PARC
  - Familiar user conceptual model (simulated desktop) --> Introduction of the "desktop" metaphor
  - Promotes recognition/pointing rather than remembering --> What You See Is What You Get" (WYSIWYG)
- Xerox Star (1981) - First commercial PC designed for "business professionals"
  - Design of the Star UI incorporated human factors as a major method of design

1980s (cont.)

- CHI conference (1982) draws 2000 - 3000 people
- Apple Lisa (1983) - successor to the Xerox Star, predecessor of the Macintosh
  - Overlapping windows
  - A commercial failure
- X Window System developed by MIT in 1984
- Apple Macintosh (1985) - a critical success

1980s (cont.)

- Emergence of new interface technologies
- Emergence of User Interface Management Systems (UIMS), toolkits, & interface builders
  - separation of the Interface from the application functions
- Emphasis on user-centered design
  - mostly preaching
- Battle between the Mac & Windows

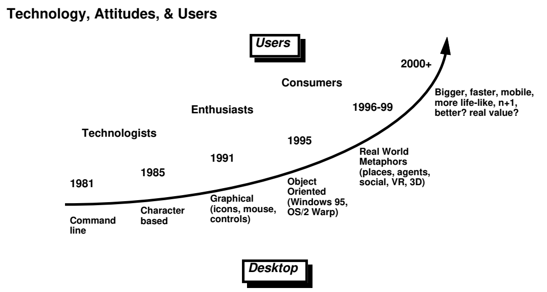
1990s

- Windows becomes king, but ...
- Growing importance and acceptance of user-centered design philosophy in industry
- Growing importance of object-oriented technologies
- Emergence of other interface modalities (e.g., speech, pen) via technological innovations, and new metaphors
- Emergence of intelligent agents --> starting to become commodity technology

1990s (cont.)

- Tremendous shift in the perception of UI design
- From a mere afterthought --> to a critical aspect of an application
- From treating users as a monolithic, homogeneous group, differentiated primarily by discipline or task --> to recognizing that users are unique

HCID Evolution



HCI Evaluation

- Literature Reviews
- User Needs Assessment
  - Determine what users need, are able to do, ...
- User Case Scenarios
  - Form of requirements analysis, used to analyze, specify, & define the system to be built
  - specifies functionality from a user's perspective

HCI Evaluation (cont.)

- Contextual Task Analysis
  - Observation/monitoring (non-invasive approach is better)
  - Interviews (exploratory, few users, subjective, structured or unstructured)
  - Surveys & Questionnaires (feedback, many users, broad sampling, highly structured)
- Simulations/Prototyping/Demos
  - Check technical feasibility, explore new ideas, EVALUATE alternatives
  - Actual working systems, or systems with simulated functionality (e.g., Wizard of Oz)

HCI Evaluation (cont.)

- Retrospective Analysis
  - User reviews own performance on a task, and provides comments
  - Provides additional insight into user's mental models
- Comparative Analysis
  - Users do the same task on multiple similar UIs or products -> Find out which one is best
- Competitive Analysis
  - Users test competitors' products, applications, & services
- Participatory Design
  - Users participate in the design of the user interface

HCI Evaluation (cont.)

- Usability Studies
  - Determine where users make errors, how often they make errors, can they use the system, number of requests for help, task completion times, etc.
  - Viewing what they do (visual), listening to what they say (auditory)
- Heuristic Evaluation
  - Use of experts and non-experts to find high-level usability problems early in the design phase

HCI Evaluation (cont.)

- Lab Studies
- Field Studies
  - Trials, site visits, on-premise structured observation, testing, and use of other data gathering techniques
- Focus Groups
  - Moderated session with few users, focused exploration & feedback

Evaluation Tools & Methods

- Wizard of Oz
  - Evaluate functions or features before developing anything, by having a human playing the role of the computer
- Rapid Iterative Developmental Testing
  - Feedback from tests of small numbers of representative users is used to suggest modifications and improvements that can be made to early design prototypes
- Failure Analysis
  - Find out where users go wrong, make mistakes, or are unsatisfied with some aspects of the design, system, etc

Evaluation Tools & Methods (cont.)

- Individual Differences Analysis
  - Determine characteristics of users who find various systems or features easy or hard to learn to use (via questionnaires, observations, or testing)
- Time Profile Analysis
  - Formalization of places and things in a user interface design where users devote the most time may reveal areas of improvement

User-Centered Design (UCD)

- Approach that focuses on users and on activities that meet users' needs.
- Embodies four key concepts:
  - Early focus on the user
  - Integrated design
  - Early and continual user testing
  - Iterative design

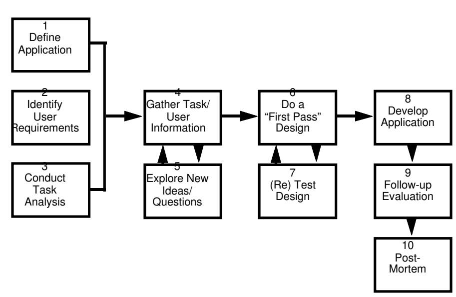
Principles of UCD

- Objective of UCD is to match whatever is being designed/developed to the characteristics
- Not much of a problem for one or two users, but problem complexity increases when there are many users
- User differences will always exist, but design for the greatest commonalties
- Focuses not on technology, but on the user's cognitive abilities, and cultural, professional, or personal preferences

Principles of UCD

- Identifies information needed from users
- Provides explicit phases for collecting and interpreting data from users
- Provides criteria for triggering moves back and forth between phases

Typical UCD Cycle



UCD Phases

- Define the Application
  Scope out the problem, and clearly lay down the ground rules.
    - What is the application?
    - Who are the intended users?
    - How and where will the application be used?
- Identify User Requirements
  Know your users, and know them well.
  Designers are not users.
  Managers & vice-presidents do not represent real users.

UCD Phases (cont.)

- Conduct a Task Analysis
  Remember, context is important.
  - What types of tasks do users typically use in order to do their jobs?
  - What cognitive, perceptual, or motor-task demands are normally imposed on users?

UCD Phases (cont.)

- Gather Existing Information on Users &Tasks in Other Applications
  Gather information; investigate where information is lacking.
  - What are the users' preferences for different interfaces?
  - What are the users' preferences for different features?
  - What factors affect usability measures (e.g., performance, satisfaction) in different interfaces?

UCD Phases (cont.)

- Explore New Ideas and Questions
  Don't be afraid to ask hard questions.
- Do a "First Pass" Design
  Prototype early. Design to clear and objectively defined usability goals.
- (Re) Test the Design
  Test repeatedly and iterate design until usability goals are met.

UCD Phases (cont.)

- Develop the Application
  By now, you should be fairly confident that you are developing the right application.
- Follow-up Evaluation
  Observe and evaluate the effectiveness of the user interface in the real world of real users. Conduct field studies.
- Post-Mortem
  PRESUME that there is a better way,and set out to find it.

Important Concepts and Terms

- contextual task analysis
- desktop
- ergonomics
- Evaluation Methods
- focus groups
- graphical user interface (GUI)
- heuristic evaluation
- human factors engineering
- human-machine interface
- input/output devices
- knowledge management
- mouse
- participatory design
- pervasive computing
- Rapid Prototyping
- simulation
- systems engineering
- task analysis
- ubiquituous computing
- usability
- use case scenarios
- User-Centered Design
- user interface design
- user requirements
- What You See Is What You Get" (WYSIWYG)
- window

Chapter Summary

- introduction to important concepts and terms
- relevance of HCID
- historical development of HCID
- emphasis on the user
  - user-centered design