Introduction to Human Computer Interaction

Jinwook Seo
Assistant Professor
School of Computer Science and Engineering
Seoul National University

Human Computer Interaction

A discipline concerned with the design, implementation, and evaluation of interactive computing systems for human use and with the study of major phenomena surrounding them.

1 - Course Introduction
HCI is interdisciplinary

Human-Computer Interaction (HCI) combines knowledge and methods associated with professionals including:

- Psychologists (incl. Experimental, Educational, and Industrial Psychologists)
- Computer Scientists
- Instructional and Graphic Designers
- Technical Writers
- Human Factors and Ergonomics Experts
- Anthropologists and Sociologists

41 BC: Emperor tired of losing the race

Win me the Chariot Race

2 - Course Introduction
Advisor intuitively finds a solution...

Hmmm......
AHA!
The Wind!

The Chariot Race

Notice aerodynamic efficiency of the faster chariot

Nuts...

Yes!!!
The Chariot Race

But, in maneuvering for position on the turn, the DRIVER makes an error!!!

Or was it the DESIGNER???

Human factors engineered

- Trade-offs between performance and usability

4 - Course Introduction
Tractors

Early design
- high center of gravity
- narrow front wheel base

Terrain
- unsurfaced and rough
- hilly

Farmer
- works long hours
- works quickly

Result

Quotes from National AG Safety Database
- older tractors have narrow front ends that are easily upset
- tractor upsets cause more fatalities than other farm accidents
- injuries often include a broken or crushed pelvis.

5 - Course Introduction
Tractors

Used to be called driver’s error

But
- accidents less frequent as modern designs have
  - roll cage
  - low center of gravity
  - wider wheel bases

So what does this teach us?

Lesson 1
- many failures of human-machine system result from designs that don’t recognize peoples’ capabilities and fallibilities
- This leads to apparent machine misuse and human error

Lesson 2
- good design always accounts for human capabilities.

How you can train yourself
- look for examples of ‘human error’
- critique them for possible ‘design error’
- propose designs that limit / remove these errors
Psychopathology of everyday things

Typical frustrations

- The engineer who founded DEC confessed at the annual meeting that he can’t figure out how to heat a cup of coffee in the company’s microwave oven.

- How many of you can program or use all aspects of your:
  - digital watch?
  - VCR?
  - car stereo?
  - washer and dryer?
  - stereo system
  - cell phones?

Remote Controls

The phone rings...

- hit pause
Remote Controls

The phone rings...
- hit pause

Why is it easier?
- big button easier to hit (Fitt’s Law)
- visually distinctive (color)
- reasonably different from other buttons
- shape and central position means its easy to find by feel in zero light conditions

TiVo designed for usability
- part of early product development

Remote Controls

But of course I’ll just learn it quickly...

six remote controls required to operate a modest home theater

cable box  digital video recorder  DVD  television  audio amplifier  VCR
Other pathological examples:

Remote control from Leitz slide projector

- How do you forward/reverse?

Instruction manual:

- short press: slide change forward
- long press: slide change backward

Still more pathological examples

Modern telephone systems

- standard number pad
- two additional buttons * and #

Problem

- many hidden functions
- operations and outcome completely invisible
  - *72+number = call forward
  - can I remember that combination?
  - if I enter it, how do I know it caught?
  - how can I remember if my phone is still forwarded?

  - Ok, I’ll read the manual
  - but what does call park mean? what’s a link?
  - where is that manual anyway?
Still more pathological examples

VCR’s, camcorders, fax machines, ...
- most people learn only basic functions
- most functionality goes untouched

The Psychopathology of computers

Britain 1976
- Motorway communication system operated 40% of it’s highways
- police controlled it in real time to
  - change lane signs, direction signs, speed limits, etc

- On December 10th, police failed to change the speed limit signs when fog descended
  - 34 vehicles crashed
  - 3 people killed
  - 11 people injured and trapped in their vehicles
  - motorway closed for 6.5 hours

10 - Course Introduction
Some quotes

Police (at inquest)
- “The system did not accept the instruction”

Dept of Transport (after examining computer logs)
- “There is no evidence of technical failure”

System designers
- after emphasizing that they have no responsibility for the system
  • “We supplied it over 5 years ago and have never been called to look at that problem”

The Coroner’s court
- judged it as "operator error"
  • the police operator:
    “failed to follow written instructions for entering the relevant data”

Where have we heard this before?

Example problems

cryptic input codes
- XR300/1: change (X) sign 300 on highway M5 (R) to code 1
- i.e. change particular sign to indicate fog condition

no feedback
- operator entered command, no visible effect of system response

cryptic error messages
- “Error code 7”

teletype machine was old, text illegible
- people could not see what they typed or system’s reply

operator overloaded with other chores
- also handled radio and telephone traffic
The PC Cup Holder

A true (?) story from a Novell NetWire SysOp

Caller: Hello, is this Tech Support?
Tech Rep: Yes, it is. How may I help you?
Caller: The cup holder on my PC is broken and I am within my warranty period. How do I go about getting that fixed?
Tech Rep: I'm sorry, but did you say a cup holder?
Caller: Yes, it's attached to the front of my computer.
Tech Rep: Please excuse me if I seem a bit stumped, it's because I am. Did you receive this as part of a promotional, at a trade show? How did you get this cup holder? Does it have any trademark on it?
Caller: It came with my computer, I don't know anything about a promotional. It just has "4X" on it.

At this point the Tech Rep had to mute the call, because he couldn't stand it.

The caller had been using the load drawer of the CD-ROM drive as a cup holder, and snapped it off the drive.

Inane Dialog Boxes

Umm, thanks for the warning, but what should I do?

What happens when you cancel a cancelled operation?

Do I have any choice in this?

Uhh... I give up on this one

12 - Course Introduction
Inane Dialog Boxes

These are too good not to show

Midwest Microwave's online catalog

13 - Course Introduction
Inane Dialog Boxes

Why should you care?

Past

- manufacturers had little incentive to emphasize usability
- customers have no experience until after they buy the product
- early technology adaptors were ‘resilient’
  - willing to put up with annoyances
- consequences of bad design typically small (annoyances)
Why should you care?

Today: Usability sells
- product reviews emphasize usability (e.g., Consumer Reports)
- customers have used related products, and can often download trial versions (including competitors)
- today’s users are impatient and intolerant of bad design

consequences of bad design now large
- costly errors in serious systems (e.g., financial institutes)
- widespread effects (e.g., incorrect billing, failures)
- life-critical systems (medical, air traffic control)
- safety (in-car navigation systems)

Why should you care?

Professionalism
- software engineers are designers
- we are ultimately responsible for the products we build
- a history of 'hack' designs does not excuse our responsibilities

Compared to civil engineers
- What would happen to an engineer who built a bridge where people fell off of it into the river (because the guard rails were too low), and where accidents were high (because the bridge was too narrow)?
- We would call this incompetence.
- The same standard should apply to software engineers.
Users and User Interfaces

Users: Person that uses something
– Not the people who create it (Users ≠ Designers)

You (=Designers) are not the user

User Interface is for interacting with users

You (=Designers) have to know the user!

User Interface is:
– Everything that users experience (UI = User Experience)
– Graphical Layout, Functionality, Navigation, Response Time

Usability

User’s view of the quality of system

How well users can use the system’s functionality

Predict and Measure
– time to learn (learnability)
– speed of performance (efficiency)
– rate of human errors
– human retention over time (memorability)
– subjective satisfaction
UI Is Important

Most users don’t read the manuals

UI affects how the system is perceived
- vital to sales
- “Usability rules the web”

Getting it right now, or pay a lot later
- 63% of large software projects go over cost
- Usability engineering saves money
- UI is a big part of software development effort (~50%)
- Bad UI can cause disasters!

Therac-25 Accidents

6 serious accidental overexposures in 19 months
Too frequent malfunctions → careless users

Aegis Combat System

Command and control system for ships

USS Vincennes shot down Iranian Air Flight 655, killing 290
July 4, 1988

Design of the Palm Beach Ballot (2000)

Confusion over Palm Beach County ballot

- 4000 people mistakenly punch the second hole
- 19000 people punch more than one hole
UI Is Important

- Can prevent disasters from happening
- Can make novices work like experts
- Can make experts more efficient
- Can reduce errors
- Can build loyal user community
  - Palm pilot, Apple iPod, iPhone

UI Is Hard To Design

- You (Designers) are not the user!
- Users request changes any time
- You can overlook tasks
- Users did not understand their own requirements

Iterative Design Process is necessary
UI Is Hard To Design

You (Designers) are not the user!

Users request changes any time
You can overlook tasks
Users did not understand their own requirements

"Only slightly more than 30% of the code developed in application software development ever gets used as intended by end-users. The reason for this statistic may be a result of developers not understanding what their users need."

Iterative Design Process is necessary
Iterative UI Design Process

Usability “engineering”

Engineer usability into systems

Iterative process

An interface design process

**Goals:**
- Articulate:
  - Who users are
  - Their key tasks

**Methods:**
- Task centered system design
- Participatory design
- User-centered design

**Products:**
- User and task descriptions
- Throw-away paper prototypes
- Testable prototypes
- Alpha/beta systems or complete specification

**Steps:**
- Brainstorm designs
- Evaluate designs
- Participatory interaction
- Task scenario walkthrough
- Graphical screen design
- Interface guidelines
- Style guides
- Usability testing
- Heuristic evaluation
- Field testing
- Completed designs

**Refined designs:**
- High fidelity prototyping methods
- Low fidelity prototyping methods

**Completed designs:**
- Alpha/beta systems or complete specification
UI Is Hard To Implement

Big part of software development effort (~50%)

Require iterative development cycle

Event processing (instantaneous response)

Prevent errors on any input (robustness)
  – undo/redo, abort

Hard to test

Foundations for designing interfaces

Understanding users and their tasks
  – Task-centered system design
    • how to develop task examples
    • how to evaluate designs through a task-centered walk-through

Designing with the user
  – User centered design and prototyping
    • methods for designing with the user
    • low and medium fidelity prototyping

  – Evaluating interfaces with users
    • the role of evaluation in interface design
    • how to observe people using systems to detect interface problems
Foundations for designing interfaces

Designing visual interfaces
- Psycholopathology/psychology of everyday things
  • what makes visual design work?
- Beyond screen design
  • representations and metaphors
- Graphical screen design
  • the placement of interface components on a screen

Principles for design
- Design principles, guidelines, and usability heuristics
  • using guidelines to design and discover usability problems

This is a great design!