What is HCI?

Sample Definitions

- "Those aspects of the system that the user comes in contact with" (1981)
- "Processes, dialogues, and actions through which a human user employs and interacts with a computer" (1987)
- "Design, evaluation, and implementation of interactive computing systems for human use and the study of major phenomena surrounding them." (1992)
- "Design of software for minds and social systems" (Suthers, 1998)
**Broadening Concerns**

- 70s: Ergonomic issues at the “man-machine interface”
- 70-80s: Psychological and information processing aspects
- 80s-90s: Workflow and workgroup
- Workspace, working practice
- Organizational issues: Impact of technologies on society

**Goals**

- Understanding, designing, evaluating and implementing interactive computing systems for human use.
- Improve the safety, utility, effectiveness, efficiency, learnability and usability of systems that include computers.
- Ensure that they integrate well in the organizational settings in which they are used.

**Why is it important?**
Usability Examples

- Stoves
- VCRs
- Car Radio (audio controls)
- Scroll Bars
- OS 10 Widgets
- WebCT: design for work practices
- Famous Examples ...

Nuclear Meltdown

Military Disaster
Flawed Elections

Consequences of Bad Interfaces

Bad interfaces may cause users to
- need more time for performing their tasks
- make more errors
- feel dissatisfied
- need more time to learn
- not learn/use the full functionality of the software
- refrain from using the software

(From Kobsa)

Things HCI Designs Can Affect

- Productivity
- Job content
- Working practices
- Job satisfaction
- Power and influence
- Physical space
Good interface design important in

- Systems with high costs of failure (e.g., nuclear power plants)
- Systems with high demands on operators (e.g., rescue coordination centers, combat aircraft, call centers)
- Mission-critical systems (e.g., space mission control)
- Systems used routinely by large numbers of people!

So how do we do it?

Factors in Design
Levels of Analysis

Examples

• (From Text):
  • Ticketing Application
  • Grocery Checkout
  • Paper Interface

Interdisciplinary

• **Computer Science**: hardware and software tools and design techniques
• **Cognitive Psychology**: behavior and underlying mental processes => capabilities and limitations of users. Perception, attention, memory, learning, thinking, problem solving.
Interdisciplinary!

- **Social and organizational psychology**: human behavior in social contexts: influences of individuals and groups on each other; relationship between group structure and group activity.
- **Ergonomics or Human Factors**: Design to fit the physiological capabilities and capacities of people. Avoiding physical damage. Applying the above disciplines to design.

Interdisciplinary!!

- **Linguistics**: Design of command languages; natural language parsing; internationalization and localization.
- **Artificial Intelligence**: Natural language and speech interfaces; computer vision; "intelligent" support based on plan recognition; automating other cognitive tasks. Designing knowledge representations for human interpreters.

Interdisciplinary!!!

- **Philosophy, Sociology and Anthropology**: Implications of introduction of IT. Ethnomethodology (basing design on observations of practice rather than predictive models). Computer Supported Cooperative Work.
- **Engineering and Design**: Empirical testing. Graphic design.
How to avoid problems for users?

- Analyze the interface using "common sense"?
- Develop a theory of "human cognitive processing" and use it to predict problems?
- Test the interface with users and watch for problems?

Measurable Factors

- Speed of performance
- Error/success rate
- Time to learn
- Retention over time
- Subjective satisfaction

Ideas about design

- Waterfall model
- Star Life Cycle
Ideas about design

- Waterfall model
- Star Life Cycle
- User-centered design
- Usage-centered design
- Participatory-design
- Integration of disciplines
- Iterative, empirical

Activities of the HCI Professional

- Evaluation
- User, work, task and environmental analyses
- Technical analyses
- Requirements specification
- Design and design representation
- Prototyping (at various levels of abstraction)
- Coding, and implementation in general