Human-Computer Interaction (HCI) Overview

• At the end of the day, it’s all about the picture below

A system’s designer/developer, must effectively communicate their perception (mental model) of the system to the system’s users through the “image” presented by that system

• Once you know how to bridge that gap, then you know HCI

based on Don Norman’s model, with some UML, software engineering, use case modeling, and MVC tweaks
How the Course is Structured

- **Theory**: concepts, principles, history, methods, techniques, models
- **Practice**: focus on specific technologies as a sandbox of sorts for testing and applying the theory
- Java Swing — knowing the libraries, designing software, implementing specific functions
- Others, depending on time: OpenGL/GLUT, HTML/CSS/JavaScript, Cocoa Framework — different technologies, similar application to HCI

How to Use the Materials

- *Designing the User Interface* by Shneiderman & Plaisant — this is the core textbook; the theory part of the course will follow the chapters in this book
- *The Design [Psychology] of Everyday Things* by Norman — basis for many core design principles; not the only one, but widely acknowledged as seminal
- *Usability Engineering* by Nielsen — basis for design methodologies; again, not the only one, but viewed as pioneering work in the area
The State of the Field

• HCI is a “young” field, when compared to other computer science topics such as computing theory, programming languages, and operating systems

• Naming: “user interface design,” “interaction architecture,” “interaction design,” “computer-human interaction”

• Related fields: cognitive psychology, software engineering, computer graphics

Characteristics of a “young” field

• Few fully established universal concepts

• “Guru-oriented” — field depends on key individuals that establish their own schools of thought; results in unstable vocabulary, jargon

• Commercial products sometimes ahead of research

Characteristics of a “mature” or “established” field

• Established, universally accepted conceptual framework

• Standardized vocabulary; individualized schools of thought shift to the cutting edge of the field

• Research leads commerce
Usability Requirements

• What do we ask of a usable system?
• An early take: the US Military Standard for Human Engineering Design Criteria (paraphrased, my italics)

1. Achieve required performance
2. Minimize skill and personnel requirements, training time
3. Achieve required reliability for personnel, equipment, software
4. Foster design standardization

Requirements are Requirements

• One view is that usability requirements are pretty much the same as software requirements in general:

1. Ascertain users’ needs
2. Ensure proper reliability
3. Promote appropriate standardization, integration, consistency, and portability
4. Complete projects on schedule and within budget
• Nielsen’s term is “system acceptability”
Usability Measures

• To imperatively determine that a requirement has been fulfilled, metrics are required — how do we quantify requirement goals (1) and (3)?

• Requirement (2): unit tests, fault tolerance, program correctness

• Requirement (4): deadlines met, products shipped, money remaining in the bank!

• International Standards Organization (ISO) standard 9241: effectiveness, efficiency, satisfaction

• Current consensus (terms in parentheses indicate Nielsen’s vocabulary):

1. Time to learn (learnability)

2. Speed of performance (efficiency)

3. Rate of errors by users (errors)

4. Retention over time (memorability)

5. Subjective satisfaction (satisfaction)

• Nielsen lists memorability as the metric #3, errors as metric #4
Usability Motivations: Not Just a Pretty Face

- *Life-critical systems*: need we say more?
- *Industry and commerce*: usability is frequently the “oil” in a well-oiled business
- *Office, home, entertainment*: Web browsers, e-mail, cell phones, and digital music players as usability successes — or are they?
- *Exploration, creation, collaboration*: traditionally very “human” activities; so far, technology gets in the way more often than not
- *Technology and society*: hanging chads, anyone?

Universal Usability: HCI as Democratizer

- *Workplace/environment diversity*: anthropometry, sometimes referred to as “human factors”
- *Human diversity*: cognitive and perceptual abilities; personality; cultural and international differences; disabilities and handicaps; age groups (older adults, children)
- *Technological diversity*: today, this deals with bandwidth (network, display), and languages (internationalization, a.k.a. “i18n”)

HCI as a Profession, Science, and Field of Study

• “Addressing the needs of all users”

• Not a “religion” — early perception of HCI as nitpickers and aesthetes, even elitist; in fact it is a full-fledged subdiscipline of computer science

  1. Growth/maturity as a field of study: seek and establish universal concepts, theory, techniques
  2. Apply the theory: develop tools to make “good HCI” easier to do — analogous to increasingly more sophisticated programming languages/compilers
  3. Raise public consciousness: unique to HCI, because it involves exactly the system/user border

Good HCI Leads to Tangible, Significant Financial Impact

• Improved software development process: time to deployment/market, cost savings, overall system performance

• Lower overhead: fewer systems/designs (ideally just one!) for a broader user base

• Enhanced competitiveness: Sony Walkman, Nokia cell phones, Apple iPod

• Expansion of markets: call centers, markets with different native languages