CIS 443/543
User Interfaces

Lecture 2: What is Usability?
(chapter 1.1-1.3)

Goals of this Course

• Learn how to design useful, usable and safe interactive software
  – Human-centered software design & development
  – Evaluation of usability
• Understand why systems and people fail to work and play together
  – Basic issues of human psychology & sociology
  – Common design flaws and how to avoid them

• What can we learn about usability from simple everyday things?
Design Lessons Learned

- Form follows function (use)
- Form follows human physical anatomy and behavior
- Form follows average or stereotyped person
- Form follows custom (culture)
- “Intuitive” interface just means the designer matched the design with what people expect!

Categories of Design for Usability

- FUNCTIONALITY PROBLEM
  - What are the functions this object can perform?
  - Will it do what I want?
- CONTROL PROBLEM
  - Which control or sequence of controls do I use to get what I want?
- FEEDBACK PROBLEM
  - How do I know I got what I wanted?
- CONTEXT PROBLEM
  - Am I using the right functions at the right time?
Key Concept: Usability

• What is usability?
  "Intuitive"?
  "Natural"?
  "User friendly"?
  "Easy to use"?
  "Idiot proof"?

• Problem
  Vague
  Subjective
  Can’t be measured or tested
  Can’t be used for design

What is usability?

• DEFINITION
  – Systematic process that develops usable systems for specific users in a specific context
  – Usability requirements + usability measures
Chapter 1.2 Usability requirements

• **DEFINITION Usability requirement**
  – Evolving detailed description of what the system should do and why with regards to human behavior
  – Does not describe how the system should do it such as the detailed system design

• Other requirements: functional, hardware, etc.

• **Usability requirements** formed early during requirements analysis phase of design

Example of a usability requirement

• Proposed system: Kiosk for rapid transit system
  – Sample Functional requirement
  – Sample Usability requirements
    • User must be able to buy a ticket
    • Blind user must be able to buy typical trip ticket unassisted in 4 minutes
  – Sample Hardware requirement
    • ATM type machine

Usability requirements analysis

• Ascertain the user’s functions
  – Determine what tasks and subtasks must be carried out (Task Analysis)
  – Task types
    • Core tasks performed frequently
    • Occasional tasks
  – Functionality must match need or else users will reject or underutilize the product
Usability requirements analysis

• Promote standardization, integration, consistency, and portability
  – Standardization: use pre-existing industry standards where they exist to aid learning and avoid errors (e.g., the W3C and ISO standards)
  – Integration: the product should be able to run within the system
  – Consistency:
    • compatibility across different product versions
    • compatibility with related paper and other non-computer based systems
    • use common action sequences, terms, units, colors, etc. within the program
  – Portability: allow user to convert data across multiple software and hardware environments

Chapter 1.3 Usability measures

• DEFINITION
  – Usability measure allows us to objectively assess the effectiveness of a usability requirement
  – Must define the target user community and class of tasks associated with the interface

Usability Measures

• 5 human factors central to evaluation:
  – Time to learn
    How long does it take for typical members of the community to learn relevant task?
  – Speed of performance
    How long does it take to perform relevant benchmarks?
  – Rate of errors by users
    How many and what kinds of errors are made during benchmark tasks?
  – Retention over time
    Frequency of use and ease of learning help make for better user retention
  – Subjective satisfaction
    User feedback via interviews, free-form comments and satisfaction scales
Usability Measures

• Other factors
  – Fatigue
  – Enjoyment
  – Safety

Usability measures (cont.)

• Usability measures are taken during UI evaluation
  – Using mockups or prototype system
  – Methods
    • Predictive methods such as average time to perform actions
    • Usability evaluation using real users

• Trade-offs in design options frequently occur between usability requirements.

• Changes to the interface in a new version may create consistency problems with the previous version, but the changes may improve the interface in other ways or introduce new needed functionality.