CS 350: Computer/Human Interaction
Lecture 23 Overview

- Compare and discuss Homework 5
- User documentation
  - Designing documentation
  - Active learning
  - Information in the interface
  - Socially mediated documentation
- **Supplemental readings**: Links on course webpage. Read for next week

CS 350: Computer/Human Interaction
Homework 5

- Nielsen guidelines for usability inspection applied to an e-commerce site
  - Use simple and natural dialog
  - Speak the user's language
  - Minimize memory load
  - Be consistent
  - Provide feedback

CS 350: Computer/Human Interaction
User Documentation

- Any stored information about how to use system
- Reference manuals, on-line help, tutorials
- Forum archives, FAQs, wikis
- Usually a diverse set of on-line and paper documents

CS 350: Computer/Human Interaction
Documentation Challenge

- How to support all documentation needs
- Novices, first-time users also need to know why as well as how; hardest to design
- Regular users need reminders, new task procedures
- Experts need most efficient procedure

CS 350: Computer/Human Interaction
Designing Documentation

- Documentation should be designed in parallel with interface design as part of the interactive process
- Need to know which tasks most people will want to know about most of the time
CS 350: Computer/Human Interaction

**Documentation and SBD**

- Develop scenarios and usability specifications that center on learning concerns
- E.g., elaborate scenarios to consider what happens if a user does not know what to do, or makes a mistake
- E.g. write sample user guides and test them in a controlled experiment

**Active Learning**

- Many novices attempt learning by doing
  - Tend not to follow step-by-step instructions
  - Tend not to plan and analyze actions
  - Try to use prior knowledge even if not applicable
  - Often make errors and trying to correct them makes things worse
- Especially true of problem domain experts with genuine goals

**Active User Paradoxes**

- Assimilation Paradox
  - People interpret new situations in terms of what they already know, but learning new tasks require going beyond what is known
- Production Paradox
  - People want to get something done, but first must spend time learning how to get something done
- Challenge is to exploit these tendencies into positives

**Systematic Documentation**

- Comprehensive hierarchical task decomposition
  - Identifies step by step instructions; gives designer's mental model of how things should work
  - Each concept is introduce, practiced, and explained
  - Prerequisite knowledge is incorporated into documents
  - Level of detail can be out of context or not match user's task

**Minimalist Instruction**

- Attempt to make documentation interactive
  - Embed training in realistic tasks
  - Allow users to get started fast without reading much
  - Rely on users to think and improvise based on previous experience
  - Anticipate and manage errors
- Requires much more study of users than just describing how software works
- Users may be confused by the incomplete and open nature of the documentation
**CS 350: Computer/Human Interaction**

**Examples of Minimalism**

- Task-oriented, but deliberately incomplete manuals
- Blocking exotic or expert functionality from novices
- Guided exploration cards that suggest interesting tasks
- Scaffolding - walk through complex process using one overall example; e.g., SBD and the case study

**Intelligent Help and Training**

- Adaptive instruction
  - Model and track knowledge held by users
  - Present new problems and activities that expand knowledge
  - Some success with math, programming
- Context-sensitive help
  - Recognize what user is trying to do
  - Software agents not very successful for general applications. E.g. MS “Clippy”
  - Wizards work well on scripted tasks

**On-line or Paper?**

- On-line documentation is ubiquitous and most software does not come with paper manuals; much cheaper for dissemination
- Large market of third-party books
- Advantages of paper?
- Disadvantages of paper?

**Information in the Interface**

- Embed instruction and information directly in interface. E.g. walk-up-and-use system like an ATM.
  - System messages: balance specificity and task relevance with length and complexity. E.g. “Error #2” vs. “Out of memory, shutting down program X”
  - Layering information: allow users who want to know to get more information
  - Tooltips can be useful if not repeating what is in interface

**Socially Mediated Documentation**

- Documentation produced by social groups
  - Users’ groups, technical forums
  - FAQs, Wiki’s
- Key issues are
  - Who organizes and maintains this documentation?
  - How to determine the validity of information?