Lecture 13
CSE 403

User oriented design

Design principles from DOET
- Provide a good conceptual model
- Make things visible
- Principle of Mapping
- Principle of Feedback

Why design matters for Software Engineering
- Software engineering is about delivering a successful software product
- Usability considerations can have a massive impact on application architecture

What is bad design of software?
- I consider software to be badly designed when I am unable to complete the task through no fault of my own
- More generally, design that impedes user performance of the task at hand

Design failures 1

Set Mode to Program,
Applications to foo.exe

Attempt 1
- Toggle mode to Program
- Attempt to enter start application
- Selected text box does not accept input

Eventual solution
- Toggle mode to Program
- Click Apply
- Enter start application
Where was the difficulty
- Visual feedback - text box looks enabled (cursor shows)
- Actual conceptual model
  - Must be in “Program Accepted mode” (which does not always coincide with entry)
- Naïve user model
  - Mode indicated by field
  - Application indicated by field
    - Ability to enter application might or might not depend on being in program mode

Launch extended desktop as external monitor (HP)
- Enable internal/external from q menu
- Display Properties->Settings
- Right click on square with 2
- Select Attached
- Select Apply

Where people have problems
- Finding the q menu
  - Internal/External?
- Finding display properties / settings
- Selection of 2
- What is “attached”?
- Confirming with apply
  - Feedback already given
    - Menu entry checked
    - Box is highlighted
- Inconsistent conceptual model – second monitor is already in use when internal/external is selected

Other design issues
- Designing for failure
- Design for multiple levels of expertise

Designing for Failure
Multiple levels of expertise

- Beginner
- Intermediate
- Expert
- In order to be successful, applications generally must cater to both intermediate and expert users
- Catering to beginners is only important in order to get them to intermediacy

Beginners

- Do not accomplish useful work with the application
- In general, do not represent a viable market
- However, it is essential for beginners to rapidly become competent
- Tutorials, scaffolding, help facilities, training wheels, . . .

Intermediates

- Core of user base
- Cannot be expected to fully remember how application works
  - Convenient rediscovery
- Stress core functionality
  - Address basic task
  - Generally, without modification

Advanced users

- Often provide business case for the application
- Stress performance and power over ease of use
  - E.g., keyboard shortcuts
- Customization and advanced features

Involving the user in design

Initial conversations with users are to understand the users’ tasks and domain, not to co-design the application with the user

Understanding user tasks

- Job application evaluation
- Pinball design
- Restaurant management