Errors & Error Assistance

Discussion Section
Wed 8:30-9:30 & Thu 1:30-2:30

Introductory HCI: User Interface Design, Prototyping, and Evaluation
CSE 440 — Autumn 2007
Role of discussion section

- Answer questions about class material
- Cover class topics in more detail
- Cover techniques useful in group projects
I DESIGNED THE USER INTERFACE MYSELF. HOW DO YOU LIKE THE COLORS?

PUKE

FLU? INTERFACE DESIGN.

Copyright © 2002 United Feature Syndicate, Inc.
YOU HAVE CHRONIC MAHJOBBIS CRAPPUS BUT THAT'S NOT WHY YOU PUKE.

HAVE YOU BEEN EXPOSED TO ANY USER INTERFACES DESIGNED BY ENGINEERS?

YES.

YOU HAVE INTERFACE POISONING. YOU'LL BE DEAD IN A WEEK.
I changed the product design.

It's worse than the old design.

You asked me to come up with a new design.

I meant a new design that's better than the old design.

Great. You could have told me that before I did all of this work.

How do you think this makes me feel?

No one would blame me for hating you.

I'm the only sane person in this company.
Overview

- Errors Happen
- Types of Error
- Strategies
  - Prevention
  - Recovery
  - Correction
- Examples
- Reference
Common physical world errors?
Common software interface errors?
Motivation

- Consider the use of an online banking system
- Compare this to interacting with human teller

Who is more graceful at coping with errors? What are the error coping mechanisms?
Deal with errors in a positive and helpful manner

- People will make errors!
- Errors we make:
  - Mistakes
  - Slips
  - False Understanding
Types of Errors

1. **Mistakes**
   - conscious deliberations that lead to an error
   - mismatch in conceptual model

2. **Slips**
   - unconscious behavior that is misdirected
   - shows up frequently in skilled behavior
   - often arises from similarities of actions

3. **False Understanding**
   - Generalizing from a single incident
Types of Slips

1. Capture error
2. Description error
3. Loss of activation
4. Mode errors
Capture error

- frequently done activity takes charge instead of one intended

- occurs when common and rarer actions have same initial sequence
  - change clothes for dinner and find oneself in bed (William James, 1890)
  - confirm saving of a file when you don’t want to replace it (the “ok” button syndrome)
  - drive to work on a Saturday morning
intended action has much in common with others that are possible

usually occurs when right and wrong objects physically near each other

- pour juice into bowl instead of glass
- go jogging, come home, throw sweaty shirt in toilet instead of laundry basket
- move file to trash instead of to folder
Loss of activation

- forgetting what the goal is while undergoing the sequence of actions
  - start going to room and forget why you are going there
  - navigating menus/dialogs and can’t remember what you are looking for
Mode Errors

- people do actions in one mode thinking they are in another
  - refer to file that’s in a different directory
  - look for commands / menu options that are not relevant
  - pen vs mouse on tablet pc
Designing for Slips

- **general rules**
  - prevent slips before they occur
  - detect and correct slips when they do occur
  - user correction through feedback and undo

- **capture errors**
  - instead of confirmation, make actions undoable
  - e.g. Mac trash can be opened and “deleted” file taken back out

- **description errors**
  - in icon-based interfaces, make sure icons are not too similar
  - check for reasonable input

- **loss of activation**
  - if system knows goal, make it explicit
  - if not, allow person to see path taken

- **mode errors**
  - have as few modes as possible (preferably none)
  - make modes highly visible
UNIX: list files with $ls$
DOS: list files with $dir$
Error: type $ls$ instead of $dir$ on windows

What kind of error is this?
(capture, description, loss of activation, mode?)

- A: mode
What can we do about errors?

- Prevent them!
- Detect them!
- Fix them!
- Make corrections inexpensive for the user (undo/redo)
Strategies: Do Nothing

- Do nothing
  - Good
    - does not chastise the participant for an error
  - Bad
    - does not provide clues for recovery
    - can be dangerous if user does not notice error
  - E.g.
    - enter letter into a numeric field (key clicks ignored)
    - put a file icon on top of another file icon (returns it to original position)
Strategies: Prevention

- Good Representation: How can it reduce errors?
Strategies: Prevention

- **Interlock**
  - Prevent users from continuing
  - Can’t move forward in a wizard if no file is selected

- **Warn**
  - warn people that an unusual situation is occurring
  - when overused, becomes an irritant
Strategies: Recovery

- Self Correction (by computer)
  - Guess the closest legal action and do it.
  - Pros: When done well can be seamless and helpful
  - Cons: Can guess the wrong thing and be obtrusive

Automatic typo correction in Word
Mediated Self Correction

- Guess a few close legal actions, and let the user choose
- Pros: Gives the user more control
- Cons
  - May not guess the correct options
  - may add to user cognitive load
- E.g.
  - Abort, Retry, Ignore
  - Compiler bringing up line of code
Strategies: Recovery

- Computer Learns
  - Ask user what they want the system to do in this situation next time
Strategies: Error Messages

- Good Error messages
  - Do not use humor
    - sometimes it’s not funny!
  - Don’t use unnecessarily severe language
    - “Symbol table full - Fatal heap error; please go buy a ram upgrade from your local Apple dealer”
    - “And the lord said, ‘lo, there shall only be case or default labels inside a switch statement”
      - Apple C compiler
Strategies: Error Messages

Good Error messages

- Use context to help the user recover
- Identify the error that triggered the breakdown
  - “Huh?” – Apple C compiler
- Do not put the blame on the user
  - “Do you want to save your file?” vs.
  - “User forgot to save file”
- Are in the user’s language (literally, but also their vocabulary)
Strategies: Error Messages

Eg: From Wai-Ling’s Research Project
Discussion

We’ve discussed errors as tools to help the user “find the right path”. What about help systems? Have any shame or fame favorites?

Many users learn by “trial and error”
  - How does error handling affect this learning style?
Reference

- Designing for Error
  - Clayton Lewis and Donald A. Norman
- Ben Bederson / Saul Greenberg