Site Design

SWE 432, Fall 2016
Design and Implementation of Software for the Web
What’s left?

• Project Reflection now out, due 12/8
• Of by one errors:
  • 2 more quizzes
  • No HW12
• Final exam
Today

• How do you help users understand if it is possible to do what they’d like to do?
• How do you help users find what they’re looking for?
• How do you organize information in a site to maximize efficiency?
Analogies: Buying a Chainsaw

- You walk into a hardware store to buy a chainsaw. What do you do?
Site design

• If users can not find what they are looking for, they will leave.
• If users take a long time to find things, your software is not usable.
• Site design considers how users interact with information, including organization, labeling, and search.
• Challenges (differences from physical world):
  • No spatial sense of scale. 50 pages? 500 pages? 50,000 pages?
  • No sense of direction. Which way did I just go?
  • No sense of location. No spatial anchoring of where I am now and how that relates to where I could go.
Planning

• Help users determine what they can do
• Support users in how they determine what to do
What can you do with this app?
Clear system task model

- Help users accomplish goals by providing clear model of how users should view system in terms of tasks
- Design to match users’ conception of high level task organization
- Help users understand what features exist and how they can be used
- Help users decompose long tasks into small pieces
- Keep task context visible to minimize memory load
Effective planning

- Help users plan most efficient ways to complete tasks
- Keep users aware of task progress, what has been done and what is left to do
- Provide constraints to avoid transaction completion slips
  - e.g., prevent users from starting task and accidentally throwing away work mid-task
Orchestration & interaction flow

• Interaction flow - the next thing the interface wants to do is exactly what user expects
  • Follow users’ mental model
  • Let user direct software
  • Keep all related tools available
• Surprises interrupt interaction flow
• Interfaces should be invisible
Interaction flow guidelines

• Don’t use dialogs to report normal behavior
• Separate commands from configuration
• Don’t ask questions, give users choices
  • Give users default input, show possible options
• Make dangerous choices hard to reach
• Design for the probable, provide for the possible
Progressive disclosure

• a.k.a. details on demand
• Separate information & commands into layers
• Present most frequently used information & commands first
Metaphors & idioms
Metaphors

• One way to communicate interaction techniques is through metaphors to the real world
Metaphors - advantages

• Leverages understanding of familiar objects & their functions
  • File cabinets, desks, telephones
• Provides **intuitive** understanding of possible affordances & eases mapping tasks to actions
  • Open a folder, throw file in trash, momentum scrolling
Metaphors - disadvantages

• Tyranny of metaphor: ties interactions closely to workings of physical world
• Adds useless overhead in extra steps, wastes visual bandwidth
• Taken literally, becomes non-sensical
  • e.g., nesting folders 10 levels deep
Alternative - Idioms

- A consistent mental model of how something works
  - e.g., Files: open / close / save / save as
- Offers intuitive understanding of affordances & interactions
- Provides consistent vocabulary for describing interactions
- Only have to learn it once
- Might have originated in real world, but thought of in terms of mental model for UI interactions
Exercise: Examples of idioms
Examples of idioms

• Email
• Clipboard: cut / copy / paste
• Format painter
• Newsfeed
• Follow item
Task Structure
Task structure

- Flow of tasks and task steps
- Task design simplicity, flexibility, efficiency
- Maintenance of locus of control
- Direct manipulation
Separate long tasks into sequences

- Reduce STM demands by having user only work on one aspect of larger task at a time
- Don’t interrupt users in the middle with unrelated tasks
- Provide closure of each subtask at the end
Design for flexibility & efficiency

- Users may take paths never envisioned by designer
- Using studies to identify different task flows, design flexible support for each
Delta: Flight Booking, New User
Delta: Flight Booking, Existing User
Anticipate likely next actions

- Based on typical observed task flows, surface options for user to take likely next steps

What if folder does not exist?

![Save As dialog]

VS.

![Folder browser]

LaToza/Bell

GMU SWE 432 Fall 2016
Keep users in control

- Important users do not feel constrained
- Want users to feel that they can do things the way they want to do them, not as software dictates to them
Navigation
Navigation usability problems

- User can’t find desired location
- User loses track of location
- User can’t remember information from another location
Navigation

- Many different contexts where navigation is important
  - Among windows & screens
  - Among panes or frames in a window
  - Among tools and menus
  - Within an information space
Information foraging

- Mathematical model describing navigation
- Analogy: animals foraging for food
  - Can forage in different patches (locations)
  - Goal is to maximize chances of finding prey while minimizing time spent in hunt
- Information foraging: navigating through an information space (patches) in order to maximize chances of finding prey (information) in minimal time
Information environment

- Information environment represented as **topology**
- Information patches connected by traversable **links**

**Examples**

- Web pages, connected by links
- Menu options & dialogs connected by commands
- Locations on map, connected by search, scroll, move interactions with map
Traversing links

• Links - connection between patch offered by the information environment

• Cues - information features associated with outgoing links from patch
  • E.g., text label on a hyperlink

• User must choose which, of all possible links to traverse, has best chance of reaching prey
Scent

- User interprets cues on links by likelihood they will reach prey
  - e.g., do I think that the “Advanced” options are likely to have the option I’m looking for?
Simplified mathematical model

• Users make choices to maximize **possibility** of reaching prey per cost of interaction

• Predators (idealized) choice = \( \max \left[ \frac{V}{C} \right] \)
  • \( V \) - value of information gain, \( C \) - cost of interaction

• Don’t usually know ground truth, have to estimate

• Predator’s desired choice = \( \max \left[ \frac{E[V]}{E[C]} \right] \)
Some design implications of information foraging theory

• Organize information into functionally related groups
• Design effective cues, describing what will be found by traversing links
• Match expectations of user’s mental model
• Provide search
Web navigation conventions
Web navigation conventions

Site ID
You are here
Local navigation
Utilities
Sections
Footer navigation
Persistent navigation

• Forms a common idiom users already understand
• Gives instant confirmation that still on the same site
• Supports consistency and standards
  • If *all* of your pages function same way, users know how to do actions & what to expect
  • Ok for specialized page like forms that are clearly different to not follow conventions.
Breadcrumbs

• Offer trail of where the user has been and how they got there
• Shows hierarchy of information space
• Shows current location
Tabs

- Example of a metaphor: tab dividers in a three ring binder or folders in a file drawer
- Partition into sections
- Advantages
  - Easily understood and self-evident
  - (Usually) hard to miss
Questions for a good site design

• Answers to the following should be obvious for a good site design
  • What site is this? (Site ID)
  • What page am I on? (Page name)
  • What are the major sections of this site? (Sections)
  • What are my options at this level? (Local navigation)
  • Where am I in the site? (“You are here” indicators)
  • How can I search?
In Class Activity: Design a course catalog & registration system

• In groups of 2 or 3
  • Design a course catalog & registration system
    • Create sketches showing key screens
    • Should support
      • browsing course catalog, registering for classes, waitlists
      • building plan of courses to take over multiple semesters to fulfill degree requirements