Sexual Misconduct, Discrimination, Harassment and/or Related Retaliation

The University of Colorado Boulder (CU Boulder) is committed to fostering a positive and welcoming learning, working, and living environment. CU Boulder will not tolerate acts of sexual misconduct (including sexual assault, exploitation, harassment, dating or domestic violence, and stalking), discrimination, and harassment by members of our community. Individuals who believe they have been subject to misconduct or retaliatory actions for reporting a concern should contact the Office of Institutional Equity and Compliance (OIEC) at 303-492-2127 or cureport@colorado.edu. Information about the OIEC, university policies, anonymous reporting, and the campus resources can be found on the OIEC website."

This image cannot currently be displayed.

Task Analysis

Today

- Tools to understand tasks
- How to group information
- Bonus: UX careers

Update on project

- Next step: Submit project ideas (by Sunday at 5pm)
- We’ll post projects on Monday
- You will bid on projects by next Friday at noon
- We’ll assign project groups before next Sunday

Big questions in user-centered design

- Who are our users? (user analysis)
- What do they do? What do they want to do? (task analysis)
- Where and how do they do it? (environmental analysis)

Tasks

- How to talk about tasks
- Task analysis
- Learning about tasks

What is a task?

What is a task?

- Something that someone does
  - Or attempts to do
- Usually a multi-step processs
- Often (in our case) involving some tool or technology

Why we analyze tasks

- Our goal in creating user interfaces is to support people in performing tasks
- So, we want to know…
  - What users' tasks are
  - How they think and talk about them
  - What steps are required

When we get it right

- Design interfaces that support tasks that are new to us

- Learn how to avoid mistakes
  - “It’s a huge pain to upload an assignment in Moodle”
  - “I can’t figure out how to join a group in Canvas”

- Optimize tasks so they can be done faster, more efficiently, with fewer errors

Talking about tasks

- Tasks have a **goal**
- Tasks may have **prerequisites**
  - Things you need to **know** or **have** to complete that task
- Tasks have a series of **steps** that the user follows

- Example: buying shoes
  - What are some possible goals?
  - What do you need to know?
  - What do you need to possess?
  - What steps do you follow?

Thinking about goals

- Goals are often more complex than they seem
  - e.g. goal of buying shoes is to get shoes
  - These options both satisfy that goal:

This image cannot currently be displayed.

This image cannot currently be displayed.

- In most cases, we need to learn more about the task in order to support it

Subtasks and hierarchical tasks

- Tasks are hierarchical, made up of subtasks
  - Example: getting a book from the library

- Tasks may have branching paths, loops, ordering
  - Go to the nearby coffee shop;
    if the line is long, go to the other one
  - Wash the dish.
    If clean, put it in the drying rack.
    If still dirty, keep washing
  - To get ready in the morning,
    brush your teeth and shower (in any order)

Task language

- It’s important that our UI reflect how users think about tasks, not just how the system sees them

This image cannot currently be displayed.

- Example: From the system view, no difference between messaging another student or instructor. But contacting the instructor is listed under **help.**

When we get it wrong

- Tasks are not described in the user's language; user can't find them
- Tasks are sequenced incorrectly; ask for the wrong information at the wrong time
  - e.g. Ask for credit card before the user chooses what to buy
- Tasks are inefficient; has extra steps

Representing tasks

- Model tasks via **hierarchical task analysis (HTA)**

- We can represent tasks as a diagram; or as pseudocode

Task models

This image cannot currently be displayed.

Tasks as pseudocode

This image cannot currently be displayed.

- The format doesn’t really matter,
  as long as we capture steps with enough detail,
  rules for navigating through plan

How to learn about tasks

- Observation
- Interviews
- Examining existing UIs and documentation
- Examining help forums

This image cannot currently be displayed.

This image cannot currently be displayed.

Learning from the internet

This image cannot currently be displayed.

This image cannot currently be displayed.

Interviewing about tasks

- Start by asking about a typical version of task:
  - “Tell me about how you made breakfast this morning.”
- Ask follow-up questions to learn more about the process
  - Is the order the same every time? Does it have to be?
  - When are things done differently? Why?
  - How are exceptions handled?

Example: class search

Example: class search

- What is the first thing we do when searching for a class?
- Multiple entry points

This image cannot currently be displayed.

What to do with task models?

Using task models

- Build interfaces that match users' view of tasks
  - UI shows common tasks
  - Includes users' language
- Use task models to assess an existing user interface
  - Via a **cognitive walkthrough**

Cognitive walkthrough

- A **heuristic** method for analyzing the usability of some system
- Tested by designers (or expert assessors) instead of typical users
- Assuming we have a good task model, we can assess a UI for its ability to support that task
- Assess the “guessability” of the UI

How to perform a cognitive walkthrough

1. Load application
2. Choose one task
3. For the first step of the task, assess the current page of the UI for the task
   - Is it obvious how to take that step?
   - Are there any misleading elements?
   - (Take notes about any potential issues)
4. Take the next step(s) and continue

Cognitive walkthrough questions

- At each step, ask:
  - Will the user try and achieve the right outcome?
  - Will the user notice that the correct action is available to them?
  - Will the user associate the correct action with the outcome they expect to achieve?
  - If the correct action is performed; will the user see that progress is being made towards their intended outcome?

This image cannot currently be displayed.

CW
worksheet

CW example

- Task: apply to CU’s Economics program
- Steps?

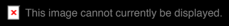
Notes from our walkthrough

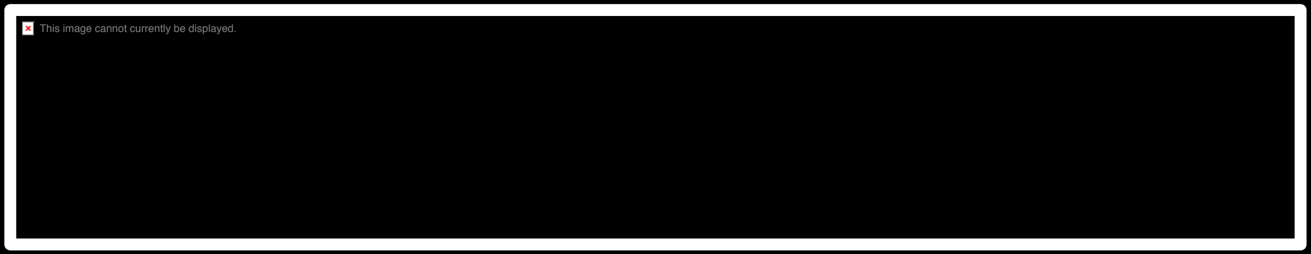
One more tool

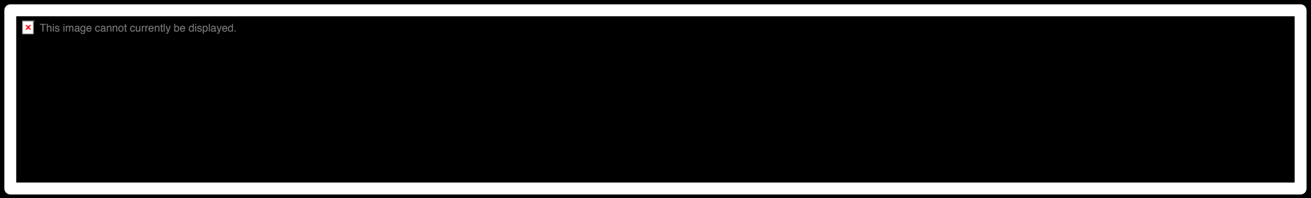
- Affinity diagramming / card sorting

The problem

- How to organize collections of static information?
- Which categories? How many?







How many menu items?

- Some “folk wisdom” about this…
- 7 items (+/- 2)
  - This corresponds to our working memory capacity
  - But little scientific evidence for this particular rule
  - Might be a good approximation anyway

Menu design strategies

- Menu items can describe the type of information

This image cannot currently be displayed.

This image cannot currently be displayed.

- … or who it's for

This image cannot currently be displayed.

- … or tasks

- Depends on specific application. What is purpose? Who uses it?

Technique: affinity diagramming

This image cannot currently be displayed.

Affinity diagram

- Useful for all kinds of things!
  - Identifying primary tasks
  - Identifying user categories about the world
  - Grouping similar functionality
  - Analyzing data from interviews, surveys

- Big idea: identify the themes and clusters in disorganized data

- Can be done by design team, or users

Affinity diagramming

1. Enumerate all of the ideas that you have; write each on a Post-It

2. Put post-its on the wall

3. Arrange into logical groups

4. Label the groups

Similar to “card sorting”

Affinity diagramming

- Let’s come up with a new mobile app for the supermarket
- Identify tasks

Practice with your neighbor

- Identify as many tasks as possible for your design (4 minutes)
- Identify 3 primary categories (4 minutes)
- If time remaining: do an HTA for one of your tasks

Next class

- Talking about users
- Project planning