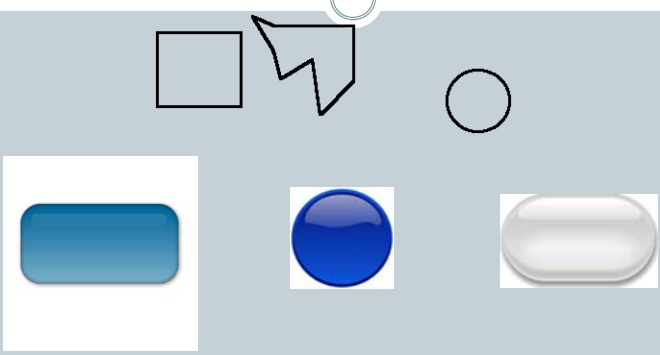
Task Analysis and Contextual Inquiry

CS160: USER INTERFACES

SETH HORRIGAN

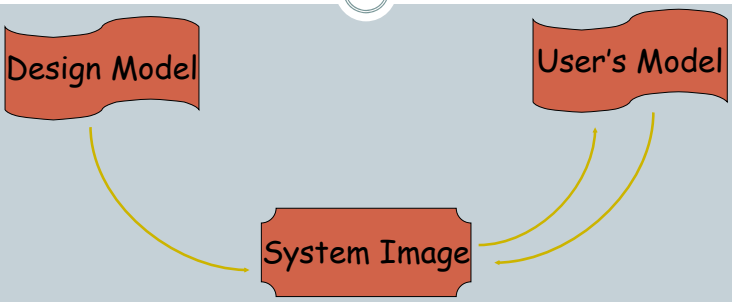
Slides based on those of John Canny, Francois Guimbretiere, Marti Hearst, James Landay, and Maneesh Agrawala

Review: Perceived Affordances



- Graphical design emphasizes affordances
- Does user recognize object as a button to be clicked?

Review: Conceptual Models



- Designers model may not match user's model
- Users get model from experience & usage
  - Users only work with system image, not with designer
- What if the two models don't match?

Review: Design Principles

1. Make controls visible

Review: Design Principles

2. Make sure mapping is clear

Mercedes Seat Adjustment

Review: Design Principles

3. Provide feedback

Due Today (before class)

Group Brainstorm
(hand in the paper copy)

Assignment (due Feb 19)

Contextual Inquiry (and Task Analysis)

- Find and interview target users (not from class)
- Analyze their tasks
- Explain how your application solves their problems
- See wiki for details

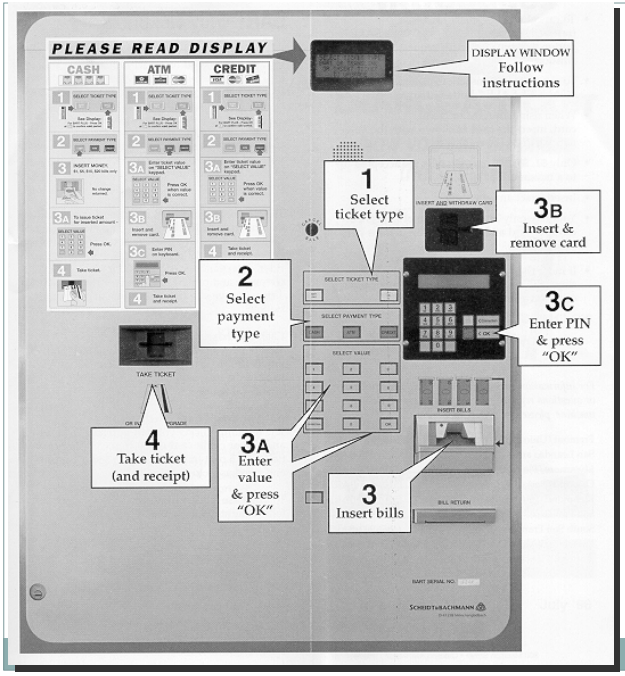
Start early – there is a lot to do

- Finding participants will take time
- We will not accept late group assignments

Topics

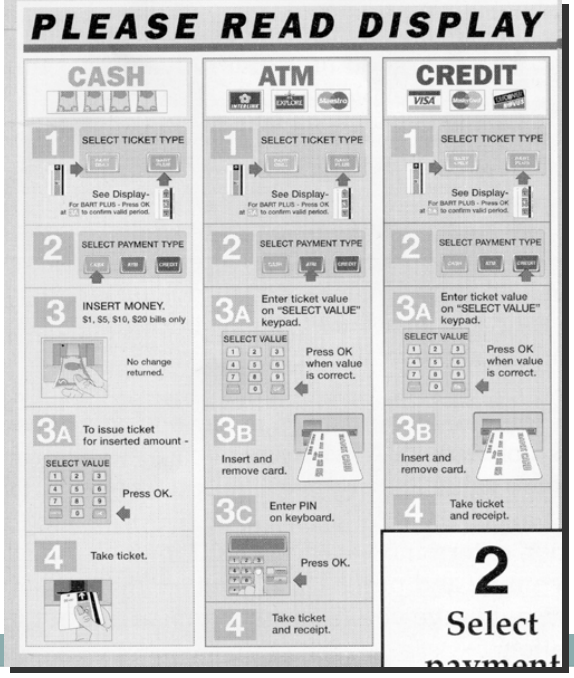
- Task Analysis
- Contextual Inquiry
- Personas

Task Analysis



Bart Ticket Machine

- Buy or add fare
- ATM, Credit, Cash



Bart Ticket Machine

- Buy or add fare
- ATM, Credit, Cash

Problems

One path of operation
- ticket type - payment type - payment - ticket

Order of payment / card insertion non-standard

Large dismiss transaction button does nothing

BART Plus has $28 min, but no indication until inserting >= $1
- can't switch to regular BART ticket

How To Improve Design?

Understand users' tasks

Designers must think about ...
- Who are the users?
- What tasks they would want to carry out?
- ...

Observe existing practices
Create scenarios of actual use
Try out ideas before building software

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

Who is going to use it?

- Identity
  - Need several typical users for broad product
- Background/Skills
  - Knowledge users already have and rely on to perform task
- Values, Likes/Dislikes
- Personal characteristics
  - Education
  - Literacy
  - Physical abilities/disabilities

    Some physical traits may be relevant: height, weight, ...
  - Age

Who (BART)?

- Identity
  - Types of users
- Background/Skills
  - Knowledge they use to perform task

Who (BART)?

- Identity
  - Tourists and visitors from elsewhere
  - Regular BART riders
    - Business people, students, disabled, elderly, etc.
- Background/Skills
  - Have an ATM card or credit card?
  - Know how to use ATM?
  - Experience with other public transit?

Who (BART)?

- Values, Likes/Dislikes
  - (i.e. May not like driving)

Who (BART)?

- Values, Likes/Dislikes
  - May not like driving
  - Want minimum fuss
  - Sometimes in a hurry
  - Maybe frugal (like saving money)
  - Maybe environmentalists
  - Hate having money eaten
  - Want to feel safe and maintain privacy
  - Hate feeling stupid

Who (BART)?

- Personal characteristics
  - Education, Physical abilities, Age, etc

Who (BART)?

- Personal characteristics
  - Mostly educated, fluent in English
  - Varying heights → don't make it too high or too low!
  - Mixture of ages, a few disabled users (e.g. wheelchairs).
  - Some bike users (make interface one-handed?)

Who is going to use it?

- Observe
  - Go out and find who uses the thing you are replacing

Task Analysis Questions

1. Who is going to use system?
2. **What tasks do they now perform?**
3. **What tasks are desired?**
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

Talk to Them

Find some real users

Talk to them

- Find out what they do now
- How would your system fit in?
- More on this a bit later

Are they too busy?

- Buy their time
  - t-shirts, coffee mugs, etc.

Old and New Tasks

Old

- The way people do things now

New

- The way you anticipate them doing things in future

Observe!

- Pick the most important tasks
- Remember you're guessing about future tasks
- Return to this when you test your prototypes

On-Line Billing Example

Dental office had billing automated

Assistants unhappy with new system

Old forms had hand-written notes

- e.g., patient A's insurance takes longer than most, etc.

B.C. CORONERS' SERVICE
DENTAL IDENTIFICATION FORM

What Tasks (BART)?

Old

- Cash to buy new ticket
- Cash to add fare to existing ticket
- Cash or credit to buy a BART Plus at window

New

- Cash, credit, or ATM card to
  - Buy new ticket
  - Add fare to existing ticket
  - Buy a BART Plus ticket

Task level of detail can vary based on goals of analysis

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. **How are the tasks learned?**
5. **Where are the tasks performed?**
6. What's the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

How are Tasks Learned?

What does the user need to know?

Do they need training?

- Book/manual information
- General knowledge / skills
- Special instruction / training

Experience, level of education and literacy

- 8th grade is often reasonable in broad design contexts

Learning Tasks (BART)

What does the user need to know?

- Walk up & use system
- Can't assume much background/training

Do they need training?

- Too time consuming

Experience, level of education and literacy

- Must be simple & similar to existing systems
- Vending machines
- ATM machines

Where is the Task Performed?

Office, laboratory, point of sale, home?

Effects of environment on users?

- Lighting, sound, comfort, interruptions, water

Social influence of environment

- Rituals, sacred places

Effects of other people (bystanders)?

- Rushing, safety, privacy

Under stress?

Where (BART)? Train Station

Where (BART)? Train Station

Loud
- Voice I/O not a good idea

Privacy
- Others can look over shoulder
- PIN must be confidential
  - Don't confirm with sound

Lighting is dim
- Make sure messages are readable

Rituals
- Panhandlers, musicians, reading the paper, cell phones

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. **What's the relationship between user & data?**
7. **What other tools does the user have?**
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

Data Relationships

Personal data
- Privacy
  - Always accessed at same machine?
  - Do users move between machines?

Common data
- Handling and processing
  - Used concurrently?
  - Passed sequentially between users?

Remote access required?

Access to data restricted?

Data Relationships (BART)

Personal data
- Users may use any machine
- Store info on BART card

Common data
- Fare rules (e.g., how much for BART Plus)
- Used concurrently

Access to data restricted?
- Only you can use your ATM or credit card

No need for remote access

Other Tools

Users work with collection of tools

- Cell phone
- Home PC
- PDA
- Timetable booklet
- Maps

Can we use other tools to facilitate interaction?

Other Tools (BART)

- Credit card, ATM card (today)
- E-wallet in cell phone or organizer (someday)
- Real-time train info on the web
- User has PC at home
  - Could provide auditing for them?
- Text on phone, use for BART delay alerts?

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the user have?
8. **How do users communicate with each other?**
9. **How often are the tasks performed?**
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

How do Users Communicate With Each Other?

Who communicates with whom?
About what?

Follow lines of the organization? Against it?

- Example: assistant to manager
  - Installation of computers changes communication between them
  - People would rather change their computer usage than their relationship

Not so relevant in context of BART

How often are the tasks performed?

Frequent users remember more details

Infrequent users may need more help

- But don't make it tedious

Which function is performed

- Most frequently? By which customers?
- Optimize system for these tasks will improve perception of good performance

Frequency (BART)?

Varying frequency of customers

- Some (most) take BART every day
- Some take it only occasionally

Varying frequency of tasks

- Can only do BART Plus every 2 weeks
- Might do add fare or buy new ticket every day
- Novices: Just one set of detailed instructions
- Experienced Users: Provide overview of process

How to find out for sure?

- Observe and interview customers!

Task Analysis Questions

1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the user have?
8. How do users communicate with each other?
9. How often are the tasks performed?
10. What are the time constraints on the tasks?
11. What happens when things go wrong?

Time Constraints

What functions will customers be in a hurry for?

Which can wait?

Is there a timing relationship between tasks?

Time Constraints (BART)?

Customers will almost always be in a hurry

Lines form

Take less than 1 minute/transaction

Be able to do any task in any order

When Things Go Wrong

How do people deal with

- Errors?
- Practical difficulties?
- Catastrophes?

Is there a backup strategy?

Things Go Wrong (BART)?

Confusion/errors on task
- "Dismiss transaction" button (that works!)

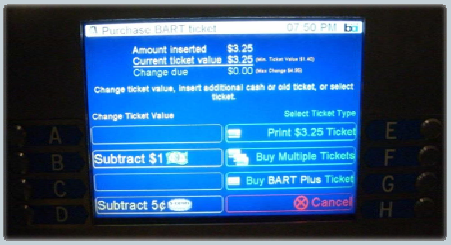
Practical difficulty
- Generated ticket with too much money
- Cash-in policy?

Catastrophe
- Machine eats card –> swipe instead of insert

Backup strategy
- Use cash in regular machines (and provide ATM)

Current State of Affairs

Current State of Affairs



Identifying Tasks

Real tasks users have faced

- Collect any necessary materials

Should provide reasonable coverage

- Compare check list of functions to tasks

Mixture of simple & complex tasks

- Easy task (common or introductory)
- Moderate task
- Difficult task (infrequent or for power users)

What Should Tasks Look Like?

- Say what user wants to do, not how user would do it
  - Allows comparing different design alternatives
- Often very specific
  - Forces us to fill out description with relevant details
    - File browser story
  - Say who the users are (use personas or profiles)
    - Design can really differ depending on who
    - Name names (allows getting more info as necessary)
    - Characteristics of the users (job, expertise, etc.)
- Some describe a complete job
  - Forces us to consider how features work together

Using Tasks in Design

- Write up a description of tasks
  - Formally or informally
  - Run by users and rest of the design team
  - Get more information where needed
- Produce scenarios covering each task
- Rough out an interface design

Using Tasks in Design

- Write up a description of tasks
- Produce scenarios covering each task
  - Three types
    - Task-based scenarios
    - Elaborated scenarios
    - Full-scale scenarios
- Rough out an interface design

Using Tasks in Design

- Write up a description of tasks
- Produce scenarios covering each task
  - Three types
    - Task-based scenarios
      - Jill is traveling to Seattle for her job next week and she wants to check on the amount she can be reimbursed for meals and other expenses.
    - Elaborated scenarios
    - Full-scale scenarios
- Rough out an interface design

Using Tasks in Design

Write up a description of tasks

Produce scenarios covering each task

- Three types
  - Task-based scenarios
  - Elaborated scenarios
    - It's Friday afternoon and Joe is flying to Sydney. He doesn't have enough money for a taxi to the airport, and he's running late.

      He goes to the local ATM and identifies himself.

      He specifies that he wants $100 from his savings account. He'd like the money in $20 notes so that he can give the taxi driver the correct change.

      He doesn't want a printed receipt, as he doesn't bother keeping track of transactions in this account.
  - Full-scale scenarios

Rough out an interface design

Using Tasks in Design

Write up a description of tasks

Produce scenarios covering each task

- Three types
  - Task-based scenarios
  - Elaborated scenarios
  - Full-scale scenarios
    - Relates your design to the tasks to be completed
    - Often done after roughing out the design

Rough out an interface design

Full-scale scenarios

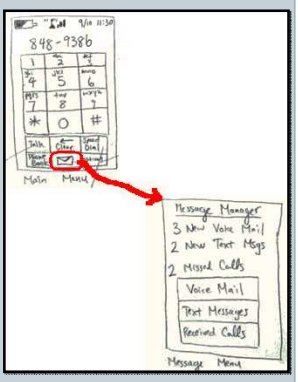
Scenarios explain how, tasks explain what

Scenarios force us to

- Show how features will work together
- Settle design arguments by seeing examples
- Only examples –> sometimes need to look beyond

Use storyboards

- sequences of sketches showing screens
- actions customers can take



Using Tasks in Design

Write up a description of tasks

Produce scenarios covering each task

Rough out an interface design

- Discard features that don't support your tasks
  - or add a real task that exercises that feature
- Major screens & functions (not too detailed)
- Hand sketched

Contextual Inquiry

Goals

- Get inside the user's head
- See their tasks the way they do
- Neither pure observation nor pure interview

Master-Apprentice Model

- Allows user to teach us what they do
  - Master (the user) works & talks
  - We interrupt to ask questions as they go
  - Each step reminds master of the next
    - Better than asking user to summarize work habits

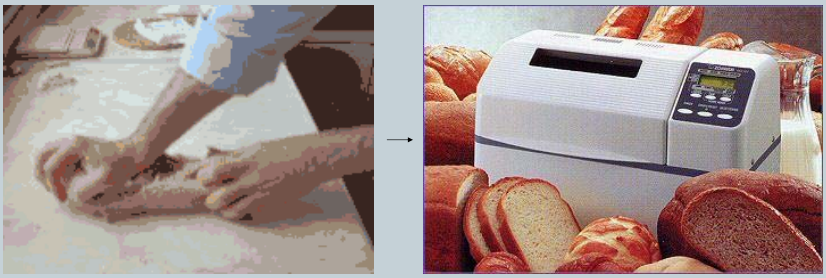
Master-Apprentice Model

- Allows user to teach us what they do
  - Skill knowledge is usually tacit (can't put it in books)
  - Sometimes literal apprenticeship is best

Master-Apprentice Model

Allows user to teach us what they do

- Skill knowledge is usually tacit (can't put it in books)
- Sometimes literal apprenticeship is best



Matsushita Home Bakery – First automatic bread maker to have twist/stretch motion [Nonaka 95]

Principles: Context

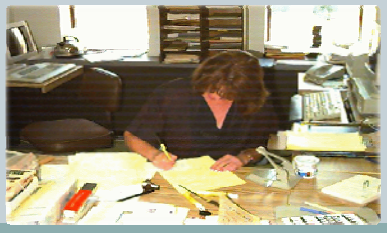
Conduct inquiry in a normal work environment

People summarize, but we want details

Keep it concrete when people start to abstract

- "We usually get reports by email", ask "Can I see one?"

Look for skipped steps, ask user to fill them in.



Show and Tell

Encourage story-telling

Principles: Partnership

Stick with master-apprentice; avoid other models, i.e.

- Avoid interviewer/interviewee
- Above all, don't "teach"!

Partnership allows more apprentice interaction

- OK to be a designer and interrupt!
- ... but go back "in role":

Alternate watching & probing (*withdrawal & return*)

Principles: Interpretation

Good facts only the starting point

- Design based on interpretations

Validate & rephrase

- Check interpretations with user
  - Users uncomfortable until phrasing is right - theirs is right by definition
- Be committed to hearing what user is really saying

Principles: Focus

You need data about specific tasks

- Steer conversation to stay on useful topics

Respect triggers (flags to change focus/understanding)

- Shift of attention (some one walks in)
- Treat every user utterance as a potential clue to ***something important***

Users: Unique or One of Many?

".. nothing any person does is done for no reason; if you think it's for no reason, you don't yet understand the point of view from which it makes sense."

"Take the attitude that nothing any person does is unique to them, it always represents an important class of customers whose needs will not be met if you don't figure out what's going on."

Thoughts on Inquiries

Establish rapport before diving in

Use recording technologies

- Notebooks, tape recorders, still & video cameras

Master/apprentice can be hard

- Staying in role – it's a lot like acting
- Don't correct! Its not a lesson!
- Its hard not designing on the fly
- Sometimes you need to put down your product

Personas

Personas (from Cooper)

"Hypothetical Archetypes"

- Archetype: (American Heritage)
  - An original model or type after which other similar things are patterned; a prototype
  - An ideal example of a type; quintessence

A precise description of user in terms

- Capabilities, inclinations, background
- Goals (not tasks)

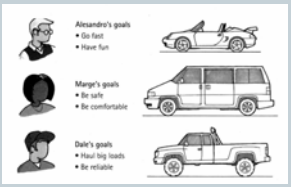
Why Personas?

Easier to generalize about specific fictional people

- We can easily discuss what Harry Potter or Scarlett O'Hara will think or do

General users have too many conflicting goals

Specific personas have clear, well-articulated goals



Defining and Using Personas

Defining them

- Identify major clusters from multiple user interviews/inquiries
- Synthesize their goals
- Check for completeness and specificity
  - Specificity prevents "elastic user"
- Try them out by developing narrative

Design each interface for a **single primary** persona

- Yet other type might use the interface

SIMS 213 Example (from Marti Hearst)

Problem Statement:

Design a new shared calendar system for UC Berkeley

Persona: Megan Richardson

- **Persona #1 - Megan Richardson**
- **Technology level: Med-low**
- **Interest in sharing events: Medium**
- **Unique situation: Currently has no calendar, would like to send events to other calendars and receive events from other calendars**

- Megan Richardson is the **22-year-old UC student** and **member of CalPirg**, the California branch of a student organization whose mission is "to deliver persistent, result-oriented public interest activism that encourages a fair, sustainable economy, and fosters responsive, democratic government." She is from Boston and has been **maintaining the CalPirg website in her spare time**. Megan created her first website as a high-school senior using Dreamweaver. She **understands basic HTML, but is not very familiar with data-driven websites or cascading style sheets**. As she has not yet worked in the business world, she has also **never used a personal calendaring system** such as Outlook.
- CalPirg sponsors **8-10 campus events each semester**, such as rallies against hunger and homelessness or for clean and affordable power. The organization attempts to **publicize** these events to its members and the general public by **posting them on their website and sending emails out to their mailing list** in order to increase attendance and catch the attention of legislators. However, because Megan is very busy with schoolwork and activism during her senior year and not many of the other CalPirg members have website design expertise, they have not had time to redesign their website in order to present their events in a coherent, easy to use, calendar-oriented format. Megan would love to have **a tool that would automate the creation of a functional, well-designed calendar** for the CalPirg website. CalPirg **might also be interested in publicizing other campus and community events that support their mission in their calendar**, as well as publicize their events in other calendars to increase attendance at their events. Megan would not want to spend more than an hour setting such a system up, and could spend only about a half hour per week maintaining information on CalPirg events. CalPirg has about 4-5 major events a semester, and 2-3 events that occur on a weekly basis. If a nicely formatted calendar could even increase attendance at their events by 10%, it would be well worth her time.
- Megan's Goals:
  - **Create a simple calendar or list of events as well as send out emails on events** that her organization sponsors on their website in order to encourage the participation of members and the public in these events without having to hire a programmer
  - To ensure that **their website supports the organization's mission**, which is to deliver persistent, result-oriented public interest activism that encourages a fair, sustainable economy, and fosters responsive, democratic government
  - **To spend most of her time on schoolwork and activism**, and less time on the technical details of managing a website

Summary

Task analysis
- Understand users and their tasks
- Real tasks with reasonable functionality coverage
- Do your best to anticipate new tasks

Contextual inquiry
- Helps answer the task analysis questions
- Hybrid between interview and observation
- Use master-apprentice model to get them to teach you

Personas
- Specific archetype of target user
- Build based on contextual inquiries/interviews

Resources

- Summary of usability methods
  - http://www.usability.gov/methods/

Next Time

Conceptual Models II

- Follow up on last Thursday's lecture

Direct Manipulation Interfaces. *User Centered System Design*. Chap 5. Hutchins, Hollan & Norman.