CS 160: Lecture 4

Professor John Canny
Spring 2003
Feb 3
Administrivia

✔ Check your team if you haven't already (outside 529 and 719 Soda).
✔ Find your team-mates after class.
Selecting Users for a new product

✔ What community are you designing for? A community is more than a set of individuals.
  * Rituals, places, shared values, background...


✔ What are your assumptions about them?
  * Make periodic “honesty checks”

✔ Seek out representative members
Selecting Tasks for Contextual Inquiry

✔ 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 the user wants to do, but not how the user would do it
  * allows comparing different design alternatives

✔ They should be very specific
  * forces us to fill out description w/ relevant details
  * 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 should 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 (us)
  * run by users and rest of the design team
  * get more information where needed

Manny is in the city at a club and would like to call his girlfriend, Sherry, to see when she will be arriving at the club. She called from a friend's house while he was on BART, so he couldn't answer the phone. He would like to check his missed calls and find the number so that he can call her back.
Using Tasks in Design (contd)

✔ 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

✔ Produce scenarios for each task
  * what user has to do & what they would see
  * step-by-step performance of task
Scenarios (cont.)

✔ Scenarios are **design specific**, tasks aren’t

✔ Scenarios force us to
  * show how various features will work together
  * settle design arguments by seeing examples
    + only examples → sometimes need to look beyond

✔ Show users storyboards
  * sequences of sketches showing screens
  * actions users can take
Involve Users to Answer Task Analysis Questions

✔ Users help designers learn
  * what is involved in their jobs
  * what tools they use
  * i.e., what they do

✔ Developers reveal technical capabilities
  * builds rapport & an idea of what is possible
  * user’s can comment on whether ideas make sense

✔ How do we do this?
  * observe & interview prospective users in work place!
Task Analysis

✔ Find out
  * who the intended customers are
  * what tasks they need to perform

✔ Observe existing work practices

✔ Create scenarios of actual use

✔ Try-out new ideas before building software
Why Task Analysis?

✔ System will fail if it
  * does not do what the customer needs
  * is inappropriate to the customer
    + “the system must match the customers’ tasks”

✔ Why not define “good” interfaces?
  * infinite variety of tasks & customers
  * guidelines are usually too vague
    + e.g., “give adequate feedback”
Example of Design Failure

✔ BART “Charge-a-Ticket” Machines
  * allow riders to buy BART tickets or add fare
  * takes ATM cards, credit cards, & cash
Please read Display

1. Select ticket type

2. Select payment type

3A. Enter value & press “OK”

3B. Insert & remove card

3C. Enter PIN & press “OK”

4. Take ticket (and receipt)
Example of Design Failure

✔ BART “Charge-a-Ticket” Machines
  * allow riders to buy BART tickets or add fare
  * takes ATM cards, credit cards, & cash

✔ Problems
  * one “path” of operation
    + ticket type -> payment type -> payment -> ticket
  * BART Plus has minimum of $28, no indication of this until after inserting >= $1
    + can’t switch to regular BART ticket
  * order of payment / card insertion non-standard
  * large dismiss transaction button does nothing
Lessons from the BART machine

✔ Failure to create convenient machine
✔ Did the designers understand/care
  * range of customers using the machine
  * what tasks they would want to carry out
  * some would find the behavior of the machine disconcerting
✔ How can we avoid similar results?
  * “What is required to perform the customer’s task?”
The Task Analysis Questions

- Who is going to use system?
- What tasks do they now perform?
- What tasks are desired?
- How are the tasks learned?
- Where are the tasks performed?
- What’s the relationship between user & data?
Questions (cont.)

✔ What other tools does the customer have?
✔ How do customers communicate with each other?
✔ How often are the tasks performed?
✔ What are the time constraints on the tasks?
✔ What happens when things go wrong?
Who?

✓ Identity?
  * in-house or specific customer is easy
  * need several typical customers for broad product

✓ Background

✓ Skills

✓ Work habits and preferences

✓ Physical characteristics
  * height?
Who (BART)?

✔ Identity?
  * people who ride BART
    + business people, students, disabled, elderly, etc.

✔ Background
  * have an ATM or credit card
  * use BART fare machines

✔ Skills
  * know how to put cards into ATM
  * know how to buy BART tickets
Who (BART cont.)?

✔ Work habits and preferences
  * not applicable

✔ Physical characteristics
  * varying heights → don’t make it too high or too low!
Talk to Them

✔ Find some real customers

✔ Talk to them
  * find out what they do
  * how would your system fit in

✔ Are they too busy?
  * buy their time
    + t-shirts, coffee mugs, etc.
What Tasks?

✔ Important for both automation & new functionality
✔ Relative importance of tasks?
✔ Observe customers
  * on-line billing example
    + small dentists office had billing automated
    + assistants were unhappy with new system
    + old forms contained hand-written margin notes
      ~ e.g., patient A’s insurance takes longer than most, etc.
What Tasks (BART)?

✔ Old tasks?
  * cash to buy new ticket
  * cash to add fare to existing ticket
  * cash or credit to buy a BART Plus at window

✔ New tasks?
  * cash, credit, or ATM card to
    + buy new ticket
    + add fare to existing ticket
    + buy a BART Plus ticket

✔ Level of detail can vary
How are Tasks Learned?

✔ What does the customer need to know?
✔ Do they need training?
  * academic
  * general knowledge / skills
  * special instruction / training
How are Tasks Learned (BART)?

✓ Walk up & use system
  * can’t assume much background/training

✓ Training?
  * too time consuming

✓ Must be simple & similar to existing systems
  * BART machines
  * ATM machines
Where is the Task Performed?

- Office, laboratory, point of sale?
- Effects of environment on customers?
- Customers under stress?
- Confidentiality required?
- Do they have wet, dirty, or slippery hands?
- Soft drinks?
- Lighting?
- Noise?
Where (BART)? Train Station

✔ Loud
  * dependence on voice I/O not a good idea

✔ Others looking over your shoulder
  * not private
  * PIN input must be confidential
    + don’t confirm with sound

✔ Lighting is dim
  * make sure messages are readable
What is the Relationship Between Customers & Data?

✔ **Personal data**
  * always accessed at same machine?
  * do customers move between machines?

✔ **Common data**
  * used concurrently?
  * passed sequentially between customers?

✔ **Remote access required?**

✔ **Access to data restricted?**
Data Relationships (BART)

✔ Personal data
  * customers 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
What Other Tools Does the Customer Have?

✔ More than just compatibility
✔ How customer works with collection of tools
  * example: automating lab data collection
    + how is data collected now?
    + by what instruments and manual procedures?
    + how is the information analyzed?
    + are the results transcribed for records or publication?
    + what media/forms are used and how are they handled?
Other Tools (BART)

✔ Not relevant
How do Customers 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 [Hersh82]
How Often do Users Perform the Tasks?

✔ Frequent customers remember more details
✔ Infrequent customers may need more help
  * even for simple operations
✔ Which function is performed
  * most frequently?
  * by which customers?
  * optimize system for these tasks will improve perception of good performance
How Often (BART)?

✔ Varying frequency of customers
  * some take BART every day (most)
  * some take it only occasionally

✔ Varying frequency of tasks
  * can only do BART Plus every 2 weeks
    + not frequent → more instructions
  * might do add fare or buy new ticket every day
    + probably more common

✔ How to find out for sure?
  * observe customers!
What are the Time Constraints on the Task?

✔ 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
What Happens When Things Go Wrong?

✔ How do people deal with
  * task-related errors?
  * practical difficulties?
  * catastrophes?

✔ Is there a backup strategy
Things Go Wrong (BART)?

✔ Confusion on task
  * “dismiss transaction” button

✔ 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 (use ATM)
Selecting Tasks

- ✔ Real tasks customers 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)
A Better Subway Machine: Hong Kong
Break
Contextual Inquiry

✔ Way of understanding users’ needs and work practices

✔ Design happens in teams
  * design team: programmer, marketing, quality assurance, producer, more..
  * user teams: the customers are also part of a team that does something
Master-Apprentice model

✔ Master - Apprentice model allows customer to teach us what they do!

* Master does the work & talks about it while working
* We interrupt to ask questions as they go
* Each step reminds the user of the next
* Skill knowledge is usually tacit (can’t put it in books)
* Studying many tasks, the designer can abstract away
* Sometimes literal apprenticeship works (Matsushita “Home Bakery”)!
Principles: Context

- Go to the workplace & see the work as it unfolds
- 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 the user to fill them in.
Principles: Partnership

✔ Stick with master-apprentice relationship; avoid lapsing into other models, i.e.
  * avoid interviewer/interviewee (stops work), expert/novice (set expectations at the start)
  * partnership allows more apprentice interaction: its OK to be a designer and interrupt!
  * ... but go back “in role”:
  * alternate between watching & probing (withdrawal & return)
Principles: interpretation

✔ Good facts are only the starting point
  * designs based on interpretations

✔ Validate & rephrase
  * run interpretations by the user to see if you are right
  * share ideas to check your reasoning (walk the chain back)
  * people will be uncomfortable until the phrasing is right
  * need to be committed to hearing what the customer is really saying (“Huh?”, “Umm…”, “Yes, but…”)
Principles: Focus

✔ Interviewer needs data about specific kind of work
  * "steer" conversation to stay on useful topics
✔ Respect triggers (flags to change focus - changing understanding)
  * shift attention (some one walks in)
  * surprises (you know it is "wrong")
  * treat every utterance by the customer 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 Interviews

✔ Use recording technologies
  * notebooks, tape recorders, still & video cameras

✔ Structure
  * conventional interview (15 minutes)
    + introduce focus & deal with ethical issues
    + get used to each other by getting summary data
  * transition (30 seconds)
    + state new rules - they work while you watch & interrupt
  * contextual interview (1-2 hours)
    + take notes, draw, be nosy! (“who was on the phone?”)
  * wrap-up (15 minutes)
    + summarize your notes & confirm what is important

✔ Master/apprentice can be hard
  * e.g., sometimes need to put down your company (the designers)
What Users Might Say

✔ “This system is too difficult”
✔ “You don’t have the steps in the order we do them”
✔ Do not take comments personally
* you shouldn’t have a personal stake
✔ Goal is to make the system easy to use for your intended users
Caveats of User-Centered Design Techniques

✔ Politics
   * “agents of change” can cause controversy
   * get a sense of the organization & bond w/ interviewee
   * important to get buy-in from all those involved

✔ Users are not always right
   * cannot anticipate new technology accurately
   * your job is to build system users will want
     + not system users *say* they want
     + be very careful about this (you are outsider)
       ~ if you can’t get users interested in your hot idea, you’re probably missing something

✔ Design forever without prototyping
   * rapid prototyping, evaluation, & iteration is key
Summary

✔ Think about the user community first
  * Who they are, what their lifestyles are, what you’re assumptions about them are.

✔ Selecting tasks
  * real tasks with reasonable functionality coverage
  * complete, specific tasks of what user wants to do

✔ Contextual inquiry
  * way to answer the task analysis questions
  * interview & observe real users
  * use the master-apprentice model to get them to teach you
Administrivia

- Meet your partners
- Let us know if your group is short of people
- Discuss your project theme
- Come up with a problem first (preferably) before you come up with a design