




# CS 160: Lecture 4

Professor John Canny

# Administrivia

---

-  You should have received your team assignment yesterday.
-  There will be time at the end of class today for you to find your team-mates and exchange contact info.
-  Make sure you start this assignment right away. There is a lot to do in the time available.

# Today's Outline

---

- 📄 **Task Analysis and Contextual Inquiry** - these are the methods you use when you first interview your target users.
- 📄 The goal is to study their behavior on specific tasks (task analysis) and learn *how users themselves understand those tasks*.






# Selecting Tasks for Contextual Inquiry

---

- ☞ Real tasks users typically face
- ☞ Should provide reasonable coverage of your target application.
- ☞ Mixture of simple & complex tasks
  - \* Easy task (common or introductory)
  - \* Moderate task
  - \* Difficult task (infrequent or for power users)

# Sample Tasks

---

-  **Easy:** Enter a street address and retrieve a map.
-  **Moderate:** Enter partial address information and search among possible locations.
-  **Hard:** Update some Point-Of-Interest information in the location database.

# What is Task Analysis ?

---

- 📄 Two flavors:
- 📄 The **formal** version breaks tasks down step by step and builds a kind of flow-chart for them. We wont be using this now...
- 📄 The **informal** version poses a set of questions to help the designer's understanding of the task. We use this one.

# The Task Analysis Questions

---

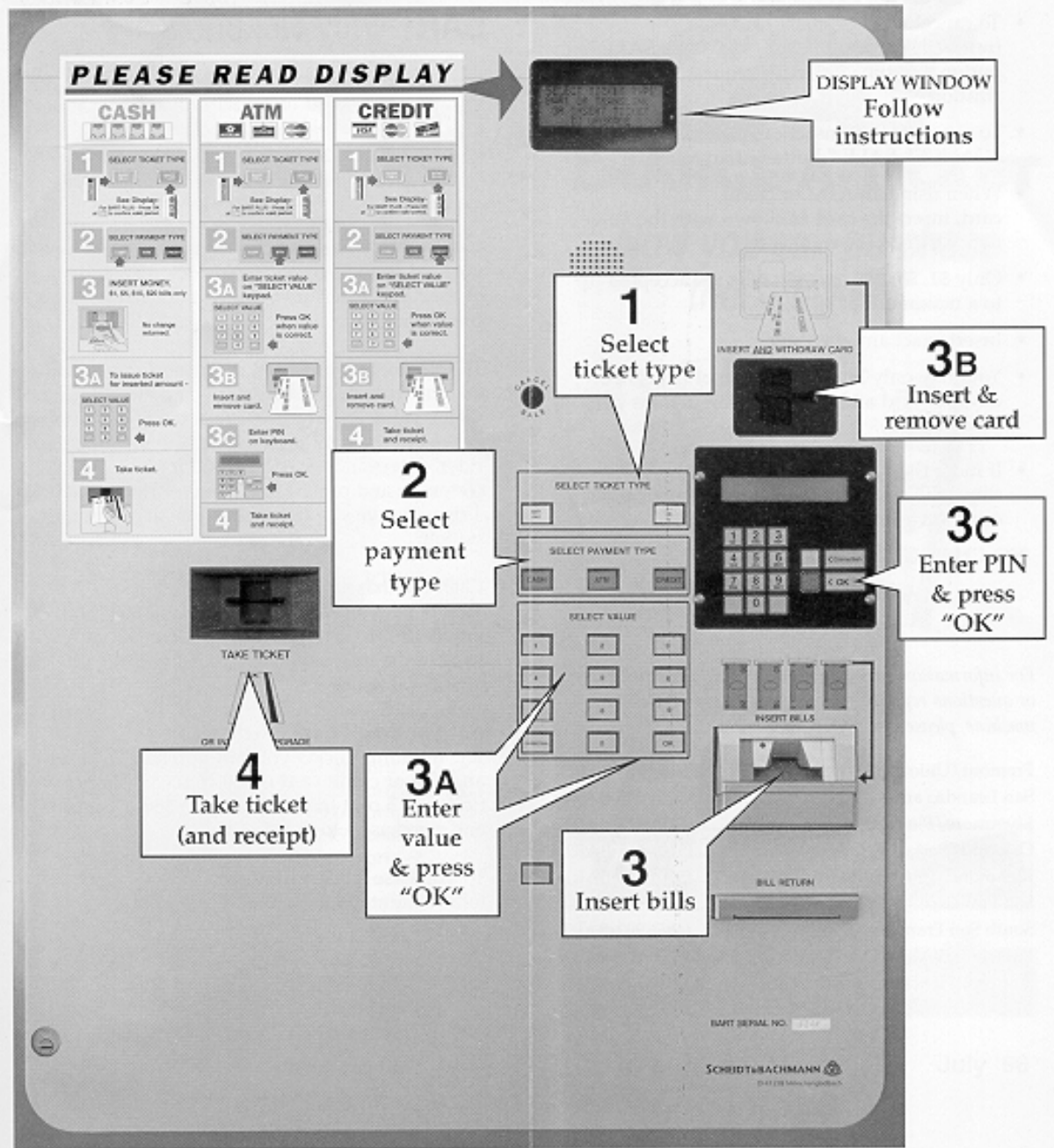
1. Who is going to use system?
2. What tasks do they now perform?
3. What tasks are desired?
4. How are the tasks learned?
5. Where are the tasks performed?
6. What's the relationship between user & data?
7. What other tools does the customer have?
8. How do customers communicate with each other?
9.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

# Example of Design Failure















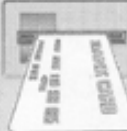


---

## BART Ticket Machines





# PLEASE READ DISPLAY

| <h2>CASH</h2>    | <h2>ATM</h2>   | <h2>CREDIT</h2>    |
|---|--|---|
| <p><b>1</b> SELECT TICKET TYPE</p>  <p>See Display -<br/>For BART PLUS - Press OK<br/>at [OK] to confirm valid period.</p> | <p><b>1</b> SELECT TICKET TYPE</p>  <p>See Display -<br/>For BART PLUS - Press OK<br/>at [OK] to confirm valid period.</p> | <p><b>1</b> SELECT TICKET TYPE</p>  <p>See Display -<br/>For BART PLUS - Press OK<br/>at [OK] to confirm valid period.</p> |
| <p><b>2</b> SELECT PAYMENT TYPE</p>    | <p><b>2</b> SELECT PAYMENT TYPE</p>    | <p><b>2</b> SELECT PAYMENT TYPE</p>    |
| <p><b>3</b> INSERT MONEY.<br/>\$1, \$5, \$10, \$20 bills only</p>  <p>No change returned.</p>                              | <p><b>3A</b> Enter ticket value on "SELECT VALUE" keypad.</p> <p>SELECT VALUE</p>  <p>Press OK when value is correct.</p>  | <p><b>3A</b> Enter ticket value on "SELECT VALUE" keypad.</p> <p>SELECT VALUE</p>  <p>Press OK when value is correct.</p>  |
| <p><b>3A</b> To issue ticket for inserted amount -</p> <p>SELECT VALUE</p>  <p>Press OK.</p>                             | <p><b>3B</b> Insert and remove card.</p>   | <p><b>3B</b> Insert and remove card.</p>    |
| <p><b>4</b> Take ticket.</p>   | <p><b>3C</b> Enter PIN on keyboard.</p>  <p>Press OK.</p>  | <p><b>4</b> Take ticket and receipt.</p>  |
| <p><b>4</b> Take ticket and receipt.</p>  | <p><b>4</b> Take ticket and receipt.</p>   | <p><b>2</b><br/>Select<br/>payment</p>  |

# Example of Design Failure

---

## BART 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  $\geq$  \$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:
  - \* What tasks customers would want to carry out?
  - \* The range of customers using the machine?
- ❏ How can we avoid similar results?
  - \* Task analysis of course...

# The 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 customer have?
8. How do customers 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?

# 1. Who?

---

 We've covered this in L1 and L2

 Values

 Likes/dislikes

 Personal characteristics:

- \* Education
- \* Literacy
- \* Physical abilities/disabilities
- \* Age

# Who (BART)?

---

## Identity?

- \* people who ride BART
  - + business people, students, disabled, elderly, etc.

## Values

- \* Broad group, generally want minimum fuss, are frugal, maybe environmentalists.

## Likes/dislikes

- \* Most people hate having their money eaten
- \* Like saving money
- \* Nervous about safety/privacy when using machines

# Who (BART cont.)?

---

## Personal characteristics

- \* Mostly educated, fluent in English
- \* Most know how to use ATM/credit card machines
- \* Most know how to buy BART tickets
- \* 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?)



## 2&3 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

# What Tasks (BART)?

---

## Old tasks?




- \* Buy new ticket with cash
- \* Add fare to existing ticket with cash
- \* Buy a BART Plus at window with cash or credit

## New tasks?

- \* Cash, credit, or ATM card for these tasks

## 4. How are Tasks Learned?

---

-  What does the customer need to know?
-  Do they need training?
  - \* Book/manual information
  - \* General knowledge / skills
  - \* Special instruction / training
-  Be careful about level of education and literacy (8<sup>th</sup> grade is a reasonable level).






# 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 (steal!)
  - \* ATM machines
  - \* Vending machines

# 5. Where is the Task Performed?

---

-  Office, laboratory, point of sale, home?
-  Effects of environment on customers?
  - \* Lighting, sound, comfort, interruptions, water
-  Social influence of environment
  - \* Rituals, sacred places
-  Effects of other people (bystanders)?
  - \* Rushing, safety, privacy
-  Customers under stress?

# Where (BART)? Train Station

---

## Loud

- \* Dependence on voice I/O not a good idea

## Not private

- \* PIN input must be confidential
  - + Don't confirm with sound

## Lighting is dim

- \* Make sure messages are readable

## Rituals:

- \* Panhandlers, musicians, reading the paper, cell phones

## 6. 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)
- \* Route maps w/ fares

## Access to data restricted?

- \* Only you can use your ATM or credit card

## No need for remote access



# 7. What Other Tools Does the Customer Have?

---

- ☞ More than just compatibility
- ☞ How user works with collection of tools, e.g.
  - \* Cell phone
  - \* Home PC
  - \* PDA
  - \* Timetable booklet
  - \* MAPs

# Other Tools (BART)

---

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

## 8. How do Users Communicate With Each Other?

---




- 📄 Who communicates with whom?
- 📄 About what?
- 📄 Follow lines of the organization? Against it?  
E.g. unions or "communities of practice"

**BART:**

- 📄 Not much. Sometimes people change bills...

# 9. How Often do Users Perform the Tasks?

---

-  Frequent customers remember more details
-  Infrequent customers 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

# How Often (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
- \* Provide a sketch of the process as well as a detailed description.
- \* Might do add fare or buy new ticket every day
- \* Just one set of instructions for novices

## How to find out for sure?

- \* Observe and interview customers!





# 10. 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

# 11. What Happens 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 (provide an ATM machine to make sure customers can get it)

# A Better Subway Machine: Hong Kong



1/30/2006

34

# Break

---


# Contextual Inquiry

---

- ☞ Way of understanding users' needs and work practices
- ☞ Goal is to "get inside the user's head" and see their tasks the way they do.
- ☞ Neither pure observation nor pure interview, but a little of both.

# Master-Apprentice model

---

 Master - Apprentice model allows user to teach us what they do

- \* Master (user) does the work & talks about it while working
- \* We interrupt to ask questions as they go
- \* Each step reminds the user of the next (better than just asking user)



# Master-Apprentice model

---

Master - Apprentice model allows user to teach us what they do

- \* Skill knowledge is usually tacit (cant put it in books)
- \* Sometimes literal apprenticeship is best: (Matsushita "Home Bakery")!



# Matsushita Home Bakery

---



1/30/2006

39

# Principles: Context

---

- 📄 Go to 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 user to fill them in.





# Principles: Partnership

- Stick with master-apprentice relationship; avoid lapsing into other models, i.e.
  - \* Avoid interviewer/interviewee (stops work)
  - \* Above all, don't "teach"!
  - \* 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

- \* Design is based on interpretations

☞ Validate & rephrase

- \* Run interpretations by user to see if you are right
- \* People will be uncomfortable until the phrasing is right - theirs is right by definition
- \* You need to be committed to hearing what the 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 - changing understanding)
  - \* Shift attention (some one walks in)
  - \* 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

## 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 (to agree with the subject)

# 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

# Summary

---

## Selecting 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 the master-apprentice model to get them to teach you