Task-Centered System Design

Lecture /slide deck produced by Saul Greenberg modified by Tony Tang

Learning Objectives

By the end of this lecture, you should be able to:

» understand how you can use task-centered design

» understand and articulate the four phases of task-centered design

» understand how to develop good task examples

The Cheap Shop Catalog Store

In Cheap Shop, people shop by browsing paper catalogs scattered around the store.

When people see an item they want, they enter its item code from the catalog onto a form.

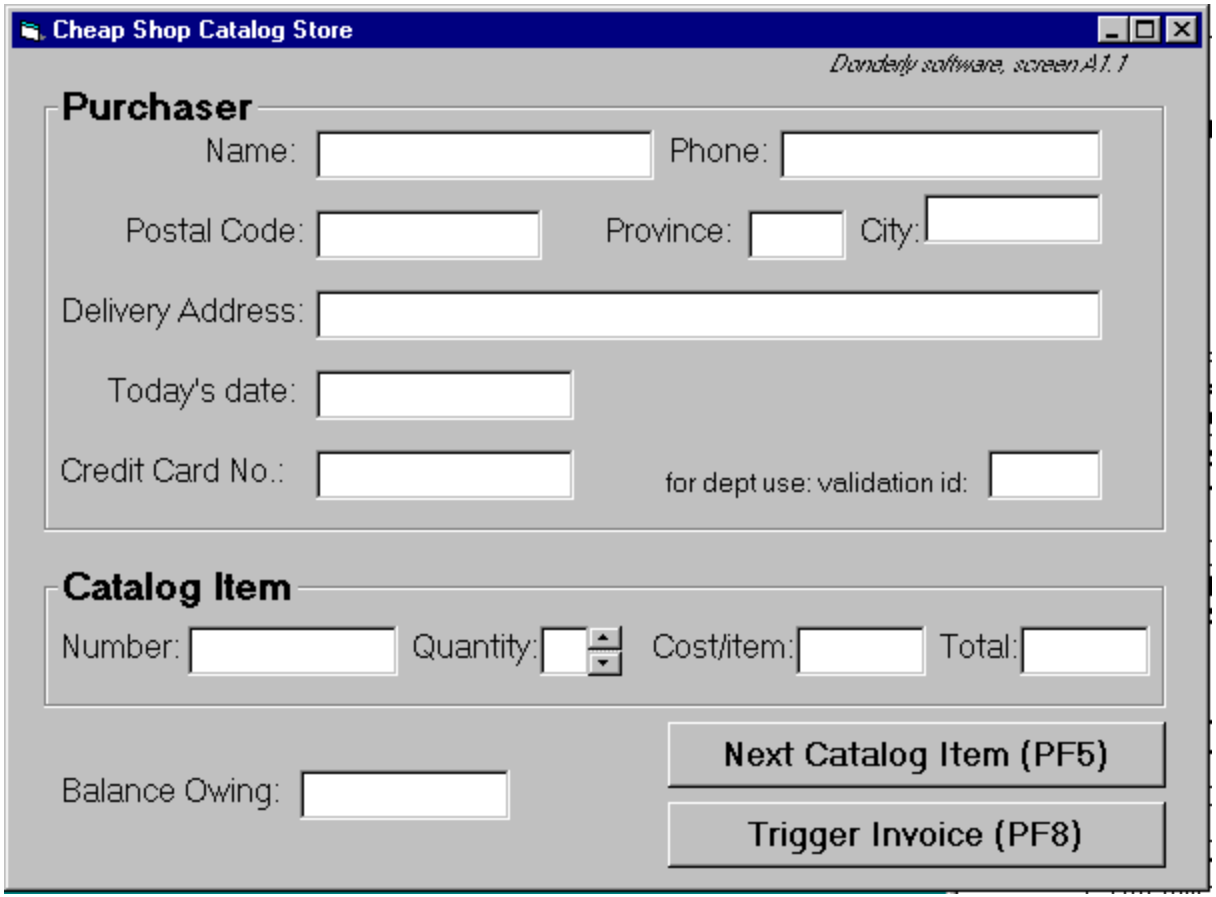
People give this form to a clerk, who brings the item(s) from the back room to the front counter.

People then pay for the items they want.

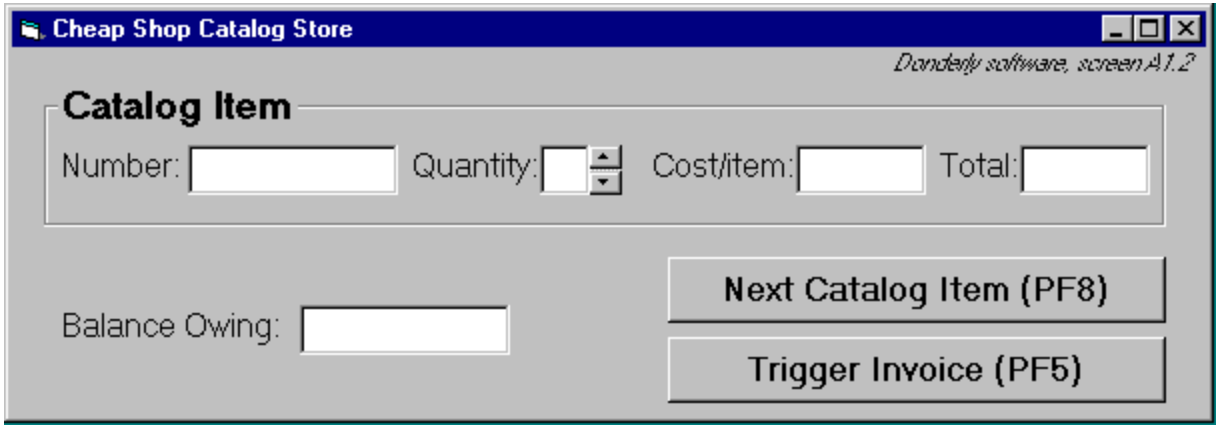
Item code	Amount
323066 697	1

Cheap Shop

Screen 1



Screen 2



Seat-of-your-pants interface design

Is cheap shop a good or bad interface?

- do you go by gut feel?
- do you go by how it looks?
- do you judge it by familiarity to other interfaces?
- if there are problems, are they minor or serious?
- did you miss anything that you really shouldn't have?
- is your opinion correct?
- how can you tell?

Alternative: are there methods where you can

- systematically determine if this interface matches the needs of its end users?
- systematically discover the usability bugs?

Requirements analysis

A software perspective

- exactly what functions should the system have?

The User

a person who will mold themselves to fit your system

Requirements analysis

An end-user's perspective

- exactly who would use the system to do exactly what?

Mary Franklin
a real person with real constraints trying to get her job done

Task Centered System Design

An end-users perspective

- exactly who would use the system to do exactly what?

Phases:

1. Identification

   identify ***specific users*** and articulate their ***concrete tasks***

2. Requirements

   decide which of these tasks and users the design will support

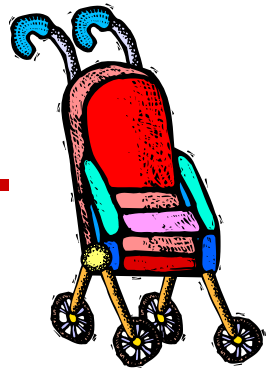
3. Design

   base design representation & dialog sequences on these tasks

4. Walkthrough Evaluations

   using your design, walk through these tasks to test the interface

Foreshadowing...



JPG Stroller. This well made but affordable Canadian stroller fits children between 1-3 years old. Its wheels roll well in light snow and mud.
…**$98.**

Red: **323 066 697**
Blue: **323 066 698**

Task example 1

- Fred Johnson, who is caring for his demanding toddler son, wants a good quality umbrella stroller (red is preferred, but blue is acceptable).
- He browses the catalog and chooses the JPG stroller (cost $98. item code 323 066 697).
- He pays for it in cash, and uses it immediately.
- Fred is a first-time customer to this store.
  He has little computer experience.
  He types very slowly with one finger.
  He lives nearby on Dear Bottom Avenue NW.

Foreshadowing...

Discussion

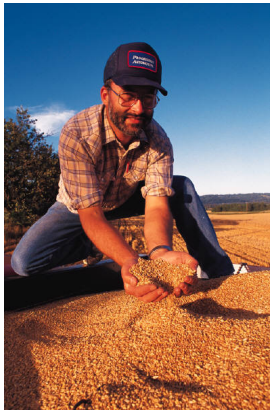
- Fred has many properties of our typical expected user:
  - many customers are first time shoppers,
  - a good number have no computer experience
  - a good number are poor typists.

- The task type is routine and important.
  - many people often purchase only one item
  - a good number of those pay by cash
  - as with Fred, people often have a general sense of what they want to buy, but decide on the actual product only after seeing what is available.

Phase 1: Identify users + tasks

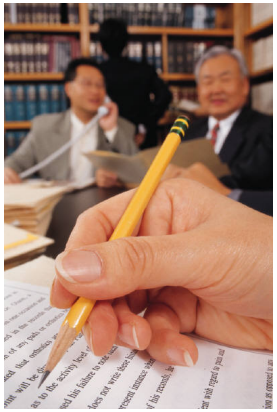
Get in touch with real people who will be potential users of your system

- prototypical categories
- extremes



Learn about their real tasks

- articulate concrete, detailed examples of tasks they perform or want to perform that your system should support
  - routine
  - infrequent but important
  - infrequent and incidental



Phase 1: Identify users + tasks

How do you identify tasks?

Immerse yourself in a real person's environment

Observe people in their actual work context

Interview people as they do their work

Shadow a person over the course of his or her day

Serve people's requests

…

Phase 1: Identify users + tasks

If there are no real users or tasks…

- think again, there probably are!

Jeff Hawkins, the inventor of the Palm Pilot, was said to have carried a small block of wood around in his shirt pocket … As various everyday situations arose, he would take out the block of wood and imagine how he would use the device.[1]

[1] *see Sato and Salvador, interactions 6(5)*

The same technique can be used to evoke a response from expected end-users

Phase 1: Identify users + tasks

If all else fails…

- describe your expected set of users,
- describe your expected set of tasks

These will become your ‘assumed users and tasks’

- verify them later as information comes in
- modify them as needed

Phase 1: Developing good task examples

1. Says what the user wants to do but does not say how they would do it
   - no assumptions made about the interface
   - can be used to compare design alternatives in a fair way

2. Are very specific
   - says exactly what the user wants to do
   - specifies actual items the user would somehow want to input

Phase 1: Developing good task examples

3. Describes a complete job
   - forces designer to consider how interface features work together
   - contrasts how information input / output flows through the dialog
     - where does information come from?
     - where does it go?
     - what has to happen next?

   Do not
     - create a list of simple things the system should do

Phase 1: Developing good task examples

4. Says who the users are
   - name names, if possible
   - says what they know

   - Why?
     - design success strongly influenced by what users know
     - can go back and ask them questions later
     - reflects real interests of real users
     - helps you find tasks that illustrate functionality in that person's real work context

Phase 1: Developing good task examples

5. Are evaluated
   - Circulate descriptions to users, and rewrite if needed
     - ask users for
       - omissions
       - corrections
       - clarifications
       - suggestions

6. As a set, identifies a broad coverage of users and task types
   - the typical 'expected' user, typical routine tasks
   - the occasional but important user, infrequent but important tasks
   - the unusual user unexpected or odd tasks

Phase 2: Requirements

Which user types will be addressed by the interface?

- designs can rarely handle everyone!
- includes why particular users are included / excluded

Which tasks will be addressed by the interface?

- designs can rarely handle all tasks
- requirements listed in terms of how they address tasks
  - Absolutely must include:
  - Should include
  - Could include:
  - Exclude:

- Discussion includes why items are in those categories

Phase 3: Design through Scenarios

Create scenarios (stories) with (thin) plots that capture the tasks that a person would engage in.

Develop designs to fit users and specific tasks

- ground interfaces in reality

Use tasks to

- get specific about possible designs (i.e. generate ideas)
- consider the real world contexts of real users
- consider how design features work together
  - what would the user do / see step-by-step when performing this task?

Phase 4: Walk-through Evaluation

Good for debugging an interface

Process

1 Select one of the task scenarios

2 For each user's step/action in the task:

a) can you build a believable story that motivates the user's actions?

b) can you rely on user's expected knowledge and training about system?

c) if you cannot:

- you've located a problem in the interface!
- note the problem, including any comments
- assume it has been repaired

d) go to the next step in the task

The Cheap Shop Catalog Store

In Cheap Shop, people shop by browsing paper catalogs scattered around the store.

When people see an item they want, they enter its item code from the catalog onto a form.

People give this form to a clerk, who brings the item(s) from the back room to the front counter.

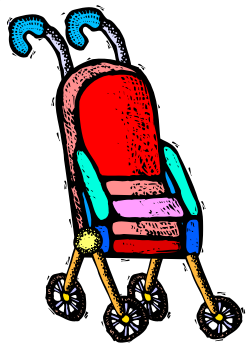
People then pay for the items they want.

Item code	Amount
323066 697	1

Developing task examples: Cheap Shop

Task example 1

- Fred Johnson, who is caring for his demanding toddler son, wants a good quality umbrella stroller (red is preferred, but blue is acceptable).
- He browses the catalog and chooses the JPG stroller (cost $98. item code 323 066 697).
- He pays for it in cash, and uses it immediately.
- Fred is a first-time customer to this store, has little computer experience, and says he types very slowly with one finger. He lives nearby on Dear Bottom Avenue NW.



JPG Stroller. This well made but affordable Canadian stroller fits children between 1-3 years old. Its wheels roll well in light snow and mud.

…**$98.**

Red: **323 066 697**

Blue: **323 066 698**

Developing task examples: Cheap Shop

Discussion

- Fred has many properties of our typical expected user:
  - many customers are first time shoppers,
  - a good number have no computer experience
  - a good number are poor typists.

- The task type is routine and important.
  - many people often purchase only one item
  - a good number of those pay by cash
  - as with Fred, people often have a general sense of what they want to buy, but decide on the actual product only after seeing what is available.

Developing task examples: Cheap Shop

Task example 2

- Mary Vornushia is price-comparing the costs of a child's bedroom set, consisting of a wooden desk, a chair, a single bed, a mattress, a bedspread, and a pillow all made by Furnons Inc.
- She takes the description and total cost away with her to check against other stores.
- Three hours later, she returns and decides to buy everything but the chair.
- She pays by credit card,
- She asks for the items to be delivered to her daughter's home at 31247 Lucinda Drive, in the basement suite at the back of the house.

- Mary is elderly and arthritic.

Developing task examples: Cheap Shop

Discussion

- Like Mary,
  - a reasonable number of store customers are elderly, with infirmities that inhibit their physical abilities.
  - a modest number of them also enjoy comparison shopping, perhaps because they have more time on their hands or because they are on low income.

- The task type is less frequent, but still important.
  - although this would be considered a 'major' purchase in terms of the total cost, the number of items purchased is not unusual.
  - delivery of large items is the norm
  - most customers pay by credit card for larger orders.

Developing task examples: Cheap Shop

Task example 3

- John Forham, the sole salesperson in the store, is given a list of 10 items by a customer who does not want to use the computer.
- The items are:
  - 4 pine chairs, 1 pine table, 6 blue place mats, 6 "lor" forks, 6 "lor" table spoons, 6 "lor" teaspoons, 6 "lor" knives, 1 "tot" tricycle, 1 red ball, 1 "silva" croquet set
- After seeing the total, the customer tells John he will take all but the silverware
- The customer then decides to add 1 blue ball to the list.
- The customer starts paying by credit card, but then decides to pay cash. The customer tells John he wants the items delivered to his home the day after tomorrow. While this is occurring, 6 other customers are waiting for John.

- John has been on staff for 1 week, and is only partway through his training program

Developing task examples: Cheap Shop

Discussion

- This task introduces the clerk as a system user.
  - Because the store has a high turnover in its staff, new employees such as John are also common.
  - Thus John reflects a 'rare' but important group of users.

- The task type is less frequent, but still important
  - The task, while complex, is fairly typical i.e., people making large numbers of purchases often ask the clerk to help them.
  - Similarly, clerks mention that customers often change their mind partway through a transaction i.e., by changing what they want to buy and/or by changing how they want to pay for it.
  - Customers, however, rarely give specific delivery dates, with most wanting delivery as soon as possible.
  - Lineups for clerks are common during busy times.

Learning Objectives

You should be able to:

» understand how you can use task-centered design

» understand and articulate the four phases of task-centered design

» understand how to develop good task examples

Next time:

» Designing scenarios and engaging in walkthroughs

Permissions

You are free:

- **to Share** — to copy, distribute and transmit the work
- **to Remix** — to adapt the work

Under the following conditions:

Attribution — You must attribute the work in the manner specified by the author (but not in any way that suggests that they endorse you or your use of the work) by citing:

> "Lecture materials by Saul Greenberg, University of Calgary, AB, Canada.
> http://saul.cpsc.ucalgary.ca/saul/pmwiki.php/HCIResources/HCILectures"

Noncommercial — You may not use this work for commercial purposes, **except** to assist one's own teaching and training within commercial organizations.

Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

With the understanding that:

Not all material have transferable rights — materials from other sources which are included here are cited

Waiver — Any of the above conditions can be **waived** if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the **public domain** under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

- Your fair dealing or **fair use** rights, or other applicable copyright exceptions and limitations;
- The author's **moral** rights;
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.