Task-Centered System Design

How to develop task examples

How to evaluate designs through a task-centered walk-through

The Cheap Shop interface

Task-Centered System Design

Requirements Analysis in HCI

• Exactly who would use the system to do exactly what?

Phase 1: Identification

• identify specific users
  - prototypical categories & extremes
• articulate example realistic tasks
  - routine
  - infrequent but important
  - infrequent and incidental

Phase 2: Design

• decide which tasks and users the design will support
• base design representation and sequences on these tasks

Phase 3: Evaluation

• walk through these tasks to test the interface
The Task-Centered Process

Get in touch with real people who will be potential users of your system
• End users
• If you can’t find them, who will actually buy/use your system?

Spend time with them discussing how the system might fit in
• Who would be willing to talk to you about this?
• Are they interested? Why or why not?

Learn about the user’s tasks
• Develop concrete, detailed examples of tasks they perform or want to perform that your system should support

Developing task examples: Cheap Shop

At Cheap Shop, people browse a catalog and then order goods from a clerk.

Task example 1:
• A man carrying a demanding toddler buys an umbrella stroller (red is preferred, but blue is acceptable), pays for it in cash, and uses it immediately.
Developing task examples: Cheap Shop

At Cheap Shop, people browse a catalog and then order goods from a clerk.

Task example 2:

• An elderly arthritic woman 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. She takes the description and total cost away with her, to check against other stores. Two hours later, she returns and decides to buy everything but the chair.

Developing task examples: Cheap Shop

At Cheap Shop, people browse a catalog and then order goods from a clerk.

Task example 3:

• A “Cheap Shop” clerk, who is 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 decides to take all but the silverware, and then adds 1 blue ball to the list.

The customer then changes his mind about paying by credit card, and decides to pay cash. The customer wants the items delivered to his home the day after tomorrow.

While this is occurring, 6 other customers are waiting for the salesperson.
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 different design alternatives in a fair way

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

3. Describes a complete job
   • not just a list of simple things the system should do!
   • does more than present a sub-goal independent of other sub-goals
   • forces designer to consider how interface features will work together
   • contrasts how information input and output is carried through the dialog
     - where does information come from?
     - where does it go?
     - what has to happen next?

4. Says who the users are
   • design success strongly influenced by what users know
   • name names, if possible
   • reflect real interests of real users
   • helps find tasks that illustrate functionality in a person’s real work context

5. Are evaluated
   • Circulate descriptions to users, and rewrite if needed
     - ask users for
       omissions
       corrections
       clarifications
       suggestions
Walk-throughs
Good for developing an interface
• debugging

Process:
1 Select one of the task scenarios
2 For each user’s step/action in the task:
• can you build a believable story that motivates the user’s actions?
• can you rely on the user’s expected knowledge and training about the system?
• if you cannot:
  - then you’ve located a problem in the interface!
  - once a problem is identified, assume it has been repaired
• go to the next step in the task

You now know
How to develop concrete task examples
How to use task examples to motivate your designs
How to evaluate designs through task-centered walk-throughs

Advanced Contextual Design
by Beyer and Holtzblatt, Morgan-Kaufmann Inc.
Reading

- Shneideman, Chapter 3, 4