

Task-Centered Design

Task-Centered Process

Creating a Task Scenario

Scenario-based Walk-throughs



Image credit: <http://worrydream.com/ABriefRantOnTheFutureOfInteractionDesign/>

Task and User Centered Design

When designing new hardware and/or software, it is important to keep (at least) two types of things in mind:

- What are the *tasks* that people need to be able to accomplish?
- Who are the *users*, what are their abilities and experiences, how will they interact with your creation to accomplish their desired tasks?

In this slide deck, we will mostly work to understand and address the first bullet...

People open these incorrectly! Why?



<https://medium.muz.li/a-design-lesson-from-a-sugar-stick-9ef86f756412>

Task-Centered System Design

Requirements Analysis in HCI

- Who would use your system?
- To accomplish what?

Phase 1: Identification

- identify specific **users**
 - “typical” categories as well as “extreme” edge cases
- articulate sample/example realistic **tasks** that they want/need to accomplish
 - everyday tasks
 - occasional yet important tasks
 - occasional and incidental and/or relatively unimportant tasks

Phase 2: Design

- decide which tasks and users your system’s design will support
- base your design representation and sequences on these tasks

Phase 3: Evaluation

- once prototyped, walk through these tasks to test the interface

Evan Golub / Ben Bederson / Saul Greenberg

The Task-Centered Process

Get in touch with *real people* who will be *potential users* of the system you are designing!

- Actual end users (not the people who manage the end users).
 - If you can’t find any, then who will actually buy/use your system?

Spend time with them discussing how the system might fit into their workflow...

- Who would be willing to talk to you about this?
- Are they interested? Why or why not?

Learn about the tasks the users want/need to accomplish!

- Develop several concrete, detailed, examples of actual or representative tasks that they perform (or want to) that your system should support...

Evan Golub / Ben Bederson / Saul Greenberg

Developing task scenarios: Super Store (I)

Imagine you have a store that plans to have a digital kiosk/catalog station where the customers browse for products, order them, and then just pick up the items from a clerk at the back of the store.

Task Scenario: A man carrying a demanding toddler buys an umbrella stroller (red is preferred but blue is acceptable), pays for it in cash, and then uses it immediately.

–What interface implication can we extract from this?

Evan Golub / Ben Bederson / Saul Greenberg

Developing task scenarios: Super Store (II)

Imagine you have a store that plans to have a digital kiosk/catalog station where the customers browse for products, order them, and then just pick up the items from a clerk at the back of the store.

Task Scenario: 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.

–What interface implication can we extract from this?

Evan Golub / Ben Bederson / Saul Greenberg

Developing task scenarios: Super Store (IIIa)

Task Scenario: The sole human clerk 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 *lör* forks, 6 *lör* table spoons, 6 *lör* teaspoons, 6 *lör* knives, 1 *lillabo* tricycle, 1 red ball, 1 “silva” croquet set. Then, after seeing the total, the customer decides to buy everything except the silverware, but then adds a blue ball to the list. The customer then changes his mind about paying by credit card, and decides to pay cash and wants the items delivered to his home the day after tomorrow...

Evan Golub / Ben Bederson / Saul Greenberg

Developing task scenarios: Super Store (IIIb)

...while all of that was occurring, 6 other customers were waiting for the clerk so they could just pick up the things they'd already bought!

–What interface implication can we extract from this?

Evan Golub / Ben Bederson / Saul Greenberg

A good task example will...

- ...say what the user wants to do, but not how they would do it. You do not want to tie any interface assumptions into the scenario description. This will allow you to compare different approaches without preconceptions.
- ...be very specific. It needs to be clear exactly what the user wants to be able to do. It needs to present what “input” the user will need to be able to make (though not necessarily how they will provide it).
- ...describe the complete task. These aren’t meant to be lists of the individual things that are done in general. We want to be able to see the big picture, not just unlinked sub-goals.
- ...identify the user. By knowing more specifics about the user, we can do things such as design based on what the user will already know how to do.
- ...be evaluated with users. Ask them if there are any things that need to be corrected or clarified, whether things have been omitted, and if there are general suggestions for additions.

Evan Golub / Ben Bederson / Saul Greenberg

Scenario-based Walk-throughs

Scenarios are good not just during designing and developing, but also for “debugging” the flow (and later perhaps the functionality) of an interface.

Example 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

Evan Golub / Ben Bederson / Saul Greenberg

What you now know about...

Developing concrete task scenarios.

Using task scenarios to motivate your designs.

Evaluating designs through task-centered walk-throughs.

Evan Golub / Ben Bederson / Saul Greenberg