

CPT208 Human-Centric Computing

Task-Centered Design

Lingyun Yu

Task-Centered System Design

How to develop task examples

How to evaluate designs via task-centered walkthroughs

Exercise: The Cheap Shop interface

Task-Centered System Design

- technique to analyze the user's tasks to inform the design of the user interface
- systematically determine if an interface matches the needs of its end users
- systematically discover usability issues



Task-Centered System Design: Example

- Cheap Shop Catalog Store
 - people browse paper catalogs in the store
 - for items they want, they fill in its catalog code into a form
 - people give the completed form to a clerk. who gets the ordered items from the back
 - people pay for the items they want



Item code	Amount
323066 697	1

Task-Centered System Design: Example

- Cheap Shop
 - wants to computerize its processes
 - develops new interface
- good or bad interface?
 - go by gut feeling?
 - go by how it looks?
 - similar to other interfaces?
 - minor or major problems?
 - missed anything?
 - your opinion correct?

The image displays two windows of the Cheap Shop Catalog Store application. The top window, titled 'Cheap Shop Catalog Store' and labeled 'Dandely software, screen A1.1', is the 'Purchaser' screen. It contains fields for Name, Phone, Postal Code, Province, City, Delivery Address, Today's date, Credit Card No., and Validation ID. The bottom window, also titled 'Cheap Shop Catalog Store' and labeled 'Dandely software, screen A1.2', is the 'Catalog Item' screen. It contains fields for Number, Quantity, Cost/item, Total, and Balance Owing, along with buttons for 'Next Catalog Item (PF5)' and 'Trigger Invoice (PF8)'.

Requirement Analysis

- software perspective
 - what functions should the system have?
 - the user:
a person who will mould themselves to fit your system
- end-user perspective
 - exactly who will use the system, to do what?
 - Esmee Elshof:
a real person with real constraints, trying to get her job done



Human-Centric Computing



Lingyun Yu

Task-Centred Design

Task-Centered System Design

- use the end-user's 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

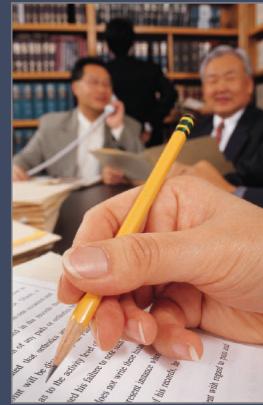
base design representations & dialog sequences of these tasks

4. Walk-Through Evaluations

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

Phase 1: Identify Users + Tasks

- get in touch with real people who will be potential users of the new system
 - prototypical categories
 - also consider extremes
- learn about their real tasks
 - articulate specific, detailed examples of tasks they perform or want to perform that the system should support
 - routine tasks
 - infrequent but important tasks
 - infrequent and incidental tasks



Phase 1: Identify Users + Tasks

- how to identify tasks?
 - **immersing** oneself into a real person's environment
 - **observing** people in their actual work context
 - **interviewing** people as they do their work
 - **shadowing** a person over the course of his or her day
 - **serving** people's requests

Phase 1: Identify Users + Tasks

- no real tasks? think again!

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. [Sato and Salvador, interactions 6(5):35–41, 1999]

such technique could also be used
to evoke responses from expected users



Phase 1: Identify Users + Tasks

- write them down as “assumed users and tasks”
 1. say what the user wants to do, not how they would do it with the interface
 - make no assumptions 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
 3. describe a complete job
 - forces designer to consider how interface features work together
 - contrasts how information input/output flows through the dialog (where does data come from, where does it go, what happens next)
 - **do not** create a list of simple things the system should do
 - **do not** present a sub-goal independent of other sub-goals

Phase 1: Identify Users + Tasks

- write them down as “assumed users and tasks”
 4. include a user description
 - name names if possible, allows to go back and ask questions later
 - say what users know, design success depends on this knowledge
 - reflects real interests of real users and helps to find tasks that illustrate functionality in that person’s real work context
 5. need to be evaluated
 - circulate to the actual users, and rewrite if necessary
 - ask for omissions, correction, clarifications, and suggestions
 6. as a set, identify a broad range of users and task types
 - the **typical user**: typical and routine tasks
 - the **occasional but important user**: infrequent but important tasks
 - the **unusual user**: unexpected or odd tasks

Phase 2: Requirements

decide which tasks & users the design will support:

- which user types will be addressed by the interface?
 - most designs will not be able to handle everybody
 - specify why particular users are included/excluded
- which tasks will be addressed by the interface?
 - most designs will not be able to handle all tasks
 - list requirements in terms of how they address tasks:
absolutely must include/should include/
could include/exclude
 - specify why tasks are in these categories

Phase 3: Design as Scenarios

- develop designs to fit users and specific tasks
 - ground interfaces in reality
- use task descriptions to
 - get specific about possible designs
 - consider real-world contexts of real users
 - consider how design features work together:
what would a user do and/or see for each step
when performing this task

Phase 4: Walk-Through Evaluation

- debug the newly developed interface design
- 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 action?
 - b) can you rely on the user's expected knowledge and training about the system to be able to perform the task?
 - c) if you cannot:
 - you have located a problem in the interface!
 - note the problem, including any comments
 - assume it has been fixed
 - d) go to the next step in the task

Back to our Example

- Cheap Shop Catalog Store
 - people browse paper catalogs in the store
 - for items they want, they fill in its catalog code into a form
 - people give the completed form to a clerk
 - the clerk gets the ordered items from the back
 - people pay for the items they want



Item code	Amount
323066 697	1

CASE ONE: GEORGE LOUKAS

Human-Centric Computing

Lingyun Yu

Task-Centered Design

Cheap Shop: Task Example 1 (Phase 1)

- George Loukas, who is caring for his demanding newborn son, wants a good quality umbrella stroller (red-blue is preferred, but blue or red are acceptable).
- He browses the catalog and chooses the JPG stroller (cost dollar \$98 item code 323 066 699).
- He pays by Alipay, and uses it immediately.
- George is a first-time customer to this store, has little computer experience, and says he types very slowly with one finger. He lives nearby in Suzhou Center.



JPG stroller. This well-made but affordable Dutch model fits children up to 6 months old. Its wheels roll well in light snow and mud. Only \$98

Red: 323 066 697
Blue: 323 066 698
Red-blue: 323 066 699

The description

Interface: Cheap Shop prototype #1.

Description #1. George Lucas, who is caring for his demanding toddler son, wants a good quality umbrella stroller (Red-blue is preferred, but red and blue are acceptable). He browses the catalog and chooses the JPG stroller (cost \$98., item code 323 066 699). He pays for it in Alipay, and uses it immediately. George is a first-time customer to this store, has little computer experience, and says he types very slowly with one finger.

Task Example 1 – Discussion (Phase 2)

- George has many properties of a typical expected user:
 - many customers are first-time shoppers in the store
 - 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
 - many of those pay by Alipay
 - as with George, people often have a general sense of what they want to buy, but decide on the actual product only after seeing what is available

CASE TWO

Human-Centric Computing

Lingyun Yu

Task-Centered Design

Cheap Shop: Task Example 2 (Phase 1)

- Yuan Zhang is price-comparing the costs of a child's bedroom set, consisting of a wooden desk, a chair, two bunk beds + mattresses, & a pillow made by Fournons Inc.
- She takes the description and total costs away with her to check against other stores.
- Three hours later, she returns and decides to buy everything but the chair.
- She pays by Master credit card.
- She asks for the items to be delivered to her daughter's apartment in Presidential Palace, on the third floor (no elevators).
- Yuan is old.



© Everything Furniture

Task Example 2 – Discussion (Phase 2)

- Like Yuan,
 - a reasonable number of customers are elderly, with limited physical abilities.
 - a modest number of customers also do/enjoy comparison shopping, perhaps because they have more time on their hands or because they are on a tight budget.
- 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; and
 - Many customers pay by credit card for large orders.

CASE THREE

Human-Centric Computing

Lingyun Yu

Task-Centered Design

Cheap Shop: Task Example 3 (Phase 1)

- Piet Jansen, the store's sole salesperson, is given a list of 10 items by a customer who doesn't 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" knifes, 1 "tot" tricycle, 1 red ball, and 1 "silva" croquet set.
- After seeing the total price, the customer tells Piet that he will take everything except 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 Piet that he wants the items delivered to his home the day after tomorrow. While this is occurring, 6 other customers are waiting for Piet.
- Piet has been on staff for only one 1 week, and is only partway though his training program.



Task Example 3 – Discussion (Phase 2)

- This task introduces the clerk as a system user.
 - Because the store has a high turnover in its staff, new employees such as Piet are also common.
 - Thus Piet 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 who buy many items often ask the clerk to help them.
 - Similarly, clerks mention that customer often change their mind partway through the transaction, i.e., by changing what they want to buy and/or how they want to pay.
 - Customers, however, rarely give specific delivery dates, with most wanting delivery as soon as possible.
 - Lineups for clerks are common during busy times.

The description

Interface: Cheap Shop prototype #1.

Description #1. George Lucas, who is caring for his demanding toddler son, wants a good quality umbrella stroller (Red-blue is preferred, but red and blue are acceptable). He browses the catalog and chooses the JPG stroller (cost \$98., item code 323 066 699). He pays for it in Alipay, and uses it immediately. George is a first-time customer to this store, has little computer experience, and says he types very slowly with one finger.

Cheap Shop: Phase 3 – Design

- **create orders:**
 - Screen 1: enter personal information and first item
 - keyboard to enter text
 - tab or mouse to switch between fields
- **further orders:**
 - Screen 2 (using *Next Catalog Item* button)
- **order completion:**
 - *Trigger Invoice*
 - shipping and billing details are printed automatically
 - system returns to a blank Screen 1
- **cancel:**
 - no input for 30 seconds
 - all input cleared and back to Screen 1
- **input checking:**
 - all input fields checked when any button is pressed
 - erroneous fields blink for 3 seconds, and then be cleared
 - shopper can re-enter correct values

The screenshot shows the Cheap Shop Catalog Store application window titled "Cheap Shop Catalog Store" with the identifier "Donders software, screenA1.1". The window is divided into two main sections: "Purchaser" and "Catalog Item".
The "Purchaser" section contains fields for Name, Phone, Postal Code, Province, City, Delivery Address, Today's date, Credit Card No., and validation id.
The "Catalog Item" section contains fields for Number, Quantity, Cost/item, Total, and Balance Owing. It also includes buttons for "Next Catalog Item (PF5)" and "Trigger Invoice (PF8)".

The screenshot shows the Cheap Shop Catalog Store application window titled "Cheap Shop Catalog Store" with the identifier "Donders software, screenA1.2". This screen is identical to screen A1.1 but lacks the "Purchaser" section.
The "Catalog Item" section contains fields for Number, Quantity, Cost/item, Total, and Balance Owing. It includes buttons for "Next Catalog Item (PF8)" and "Trigger Invoice (PF5)".

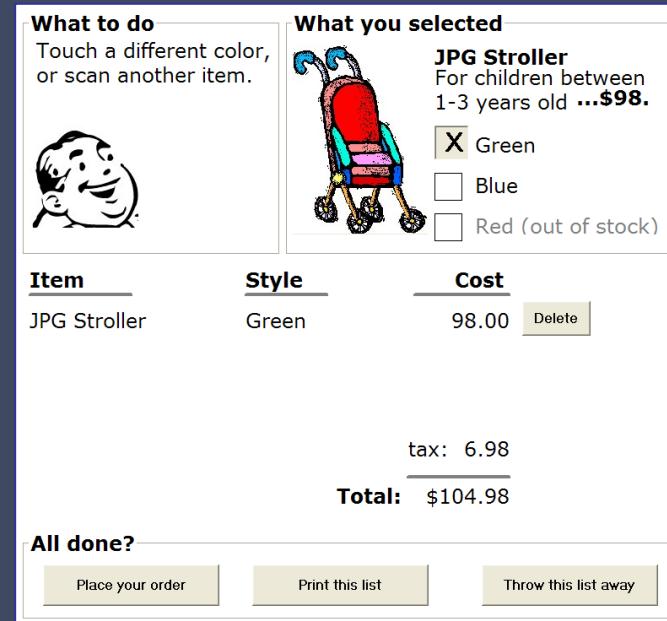
Cheap Shop: Phase 4 – Evaluation

- Walkthrough template

Description of Step	Does the user have the knowledge/training to do this?	Is it believable that they would do it? Are they motivated?	Comment / solution

Walk-Through Evaluations: Better Ways

- task-centered prototypes:
e.g., partial wizard
approaches to tasks
(explain what to do next)
- other types of prototypes
 - paper prototype:
45 minutes
 - scripted animation:
2 hours
- do these work?
do a task-centered walk-trough to find out!



IN CLASS ACTIVITY

Human-Centric Computing

Lingyun Yu

Task-Centered Design

Group activity

- Work with your group
- Two options:
 - design an innovative system that can support travelers to record the places that they travelled, the pictures/tweets that they made in each place
 - design an intuitive interaction method / interactive system to design / play LEGO

Method

- **Step 1.** generating a list of expected users, and an initial list of tasks.
Your goal is to generate an initial list of concrete task descriptions.
- **Step 2. validating the tasks.** The next step is to get a reality check of your task list. Have end-users and/or client representatives review your tasks.
- **Step 3. deciding upon key users and a tentative list of requirements.**
The task examples will provide clues on specific system requirements that you could use to design your system as well as who your target users will be.
- **Step 4. sketch the design of a new system** that addresses the users' needs (be creative and try different ideas)
- **Step 5. develop a low-fidelity prototype of the selected design – next week (March 26 Seminar session, take papers, colorful pencils, sticky notes, tapes and scissors with you)**

Summary

- task-centered design
 - how to develop concrete task examples
 - how to use task examples to motivate designs
 - how to evaluate designs through task-centered walk-throughs

References

- Task-Centered Design:
 - [Working through Task-Centered System Design](#). Greenberg, S. (2003) in Diaper, D. and Stanton, N. (Eds) *The Handbook of Task Analysis for Human-Computer Interaction*. Lawrence Erlbaum Associates.

An Interface Design Process

