Introduction to HCI

Mental and Conceptual Models

Prof. Narges Mahyar
UMass Amherst

nmahyar@cs.umass.edu
Courses, projects, papers, and more: http://groups.cs.umass.edu/nmahyar/

© Mahyar with acknowledgements to Joanna McGrenere and Leila Aflatoony
Today

• Discussion of readings [10 min]

• Mental models [40 min]
  • Conceptual models
  • Conceptual design

• In class activity [20 min]
  • Conceptual design
Discussion on requirement readings [10 min]

• Discuss with your team the following questions [3 min]
  • What you learned?
  • What surprised you?
  • How can you use this knowledge in your project?

• A randomly assigned team will present results to the class [7 min]
Why look at cognition?

Part of doing good design is understanding how people reason and react to interface experiences.

Cognitive frameworks: help us do this!
- theories of cognition provide predictive and explanatory power for understanding user behaviour.

- **Internal frameworks:**
  - the mental process inside users head

- **External frameworks:**
  - interactions with technologies, environment, context
Mental models

• "In interacting with the environment, with others, and with the artifacts of technology, people form internal, mental models of themselves and of the things with which they are interacting."

-Norman (in gentner & stevens, 1983)

• People use their mental models to:
  • Reason about a system
    - How to interact with it; how it works
  • Figure out what to do when things go wrong
Mental models vs. Conceptual models/design

**Mental models: something the user has (forms)**
- Users “see” the system through mental models
- Users rely on mental models during usage
- There are various forms of mental models
- Mental models can support users’ interaction

**Conceptual models and conceptual design**
- This is what the designer does, to foster good mental model formation by the user.
Introduction to Conceptual Models

https://www.youtube.com/watch?v=pAOyWFOFhsg
The Designer’s Model, the User’s Model, and the System Image

the user also has a mental model. they don’t necessarily match.
Recall our design concepts:

**the basics:**
(elements of these in many of the others)

- affordance
- signifiers
- mapping
- constraints
- feedback

**other concepts:**

- findability
- transfer effects
- cultural associations
- individual differences

→ all inform a user’ mental model
An object that helps you form a mental model: scissors

• Affordances:
  • Holes for something to be inserted

• Constraints:
  • Big hole for several fingers, small hole for thumb

• Mapping:
  • Holes-for-fingers suggested / constrained by appearance

• Positive transfer and cultural idioms
  • Learnt when young; constant mechanism

• Mental model:
  • Physical object implies how the operating parts work

A reasonable mental model can be formed by just looking at and perhaps holding the object.
• Some things you don’t understand you do anyway: why big blade down?
An object that **hinders** mental model formation: “old style” digital watch

- **Affordances - mixed:**
  - Four buttons are clearly for pushing and the screen shows a number – but unclear what the *entire object* affords
    
    *telling time? Setting alarms, timers, viewing heartrate, other data?*

- **Visibility – lousy:**
  - What will happen if you push each button? What mode is watch in?

- **Constraints and mapping - unknown:**
  - No visible relation between buttons, possible actions and end result

- **Transfer of training:**
  - Little relation to analog watches. But, maybe from other digital devices.

- **Cultural idiom:**
  - Some standardized core controls and functions but others variable

- **Mental model:**
  - Must be taught, or learned by trial/error
Norman’s seven-stage model

What mental models tell the **user**

What do I want to do with the system next?

**establishing goals**

Intention to act

Sequence of actions

Execution of the action sequence

What can I do next?

What if I do this?

To do it, I’ll do this, then this

**Evaluation of interpretations**

Interpreting the perception

Perceiving the state of the world

What did I do to make the system do that?

What did the system just do?

What am I now seeing?

What will I see as a result?

What do I want to do with the system next?

**the world**

What if I do this?

What can I do next?
Norman’s seven-stage model
what is it good for?

• Internal framework: best for exploratory learning
  • But this is just one way to form a mental model of a system

• Less applicable to highly learned, semiautomatic behavior
  • User has already developed strong expectation of what will happen/how it will happen
  • Gulfs in these cases tend to be very small (scissors example)
Acquiring mental models

• **During system usage:**
  - The user’s own activity leads to a mental model
  - Explanatory theory, developed by the user
  - Often used to predict future behavior of the system

• **Observing others using the system:**
  - Casual observation of others working
  - Asking someone else to “do this for me”
  - Formal training sessions

• **Reading about a system**
  - Documentation, help screens

• This is done **by the user** (not the designer)
Some characteristics of mental models

- Incomplete
- Constantly evolving
- Not accurate representation
  - (contain errors and uncertainty measures)
- Provide a simple representation of a complex phenomena
- Can be represented by a set of if-then-else rules
Conceptual Models & Conceptual Design
Conceptual models & conceptual design:

• Conceptual models describe how an interactive system is organized
  • The user also has a mental model. They don’t necessarily match.

• Conceptual model = the foundation of the interface.
  • Different user interfaces could be built upon it
  • There are many ways to represent a conceptual model

• Goal of conceptual design, how do conceptual models fit?
  • Interface design translates the cm into things we can see and interact with. It involves design choices, but must stay faithful to the concepts and terminology of the CM.
What is conceptual design?

• Crossing the gap from **requirements** to a **solution**

• Starts with brainstorming; multiple iterations to narrow down

• A conceptual design
  - Can take many different forms
  - Be built through many approaches
  - Is essentially a **set of ideas**
Mental models vs. conceptual models

• **Mental models**: *something the user has (forms)*
  - Users “see” the system through their own mental models
  - Users **rely** on mental models during usage
  - There are various **forms** of mental models
  - Mental models can **support** or **impede** users’ interaction

• **Conceptual models**: *articulation of designer’s (i.e. Your) mental model*
  - What users **will be able to do**
  - What **concepts or knowledge** users will need, in order to interact
  - **How they will** interact with system (at a very high level)
Big Picture – WHEN do task examples and requirements happen?

**GOALS**
- Understand USERS:
  - who they are
  - their key tasks

**MATERIALS / METHODS**
- Understand DESIGN:
  - design space and risks
  - choose design approach

**PRODUCTS**
- Make use of:
  - requirements
  - task examples, analysis
  - real & virtualized users
  - technology options
  - company IP

- Evaluate w/:
  - observation
  - interview/quest
  - participatory interaction
  - task walk-throughs

- Low fidelity prototyping methods

**PRE DESIGN**
- user and task descriptions
- design requirements

**EARLY DESIGN**
- throw-away prototypes
- design direction
- risk analysis

**MID DESIGN**
- testable medium-fidelity prototypes

**LATE DESIGN**
- alpha/beta systems or
  - complete specification

**CONFIRM & debug:**
- performance in real use

- Field testing

K MacLean - derived from version by Saul Greenberg (U Calgary)
A conceptual model excludes

• Low level presentation
• Implementation details
• Menu and screen designs
• Widgets
• Etc.

if you started here, you will probably get into trouble
A conceptual model can include:

- **Any central design metaphors** and analogies
  e.g. The “desktop metaphor”

- **Concepts** – objects, actions you can do to them; user roles; attributes of both.
  E.G., Files and folders; both can be opened, have names;

- **Relationships** among concepts
  e.g., Files are contained in folders

- **Mappings** from concepts to the user experience envisioned;
  e.g., The users can browse files, and mark favorites

- **Terminology** that will be used (consistently) to tie it all together

- **Interaction** types; how will they interact with it?
  E.G. Give commands, perform operations, explore

- **Interface** types; is it/should it be constrained? How would different interfaces affect result?
Metaphors

• Well known concepts you can rely on to help users understand and interact with the system

• Many kinds, e.g.,

• Interactions
  • Swipe to turn page in an ebook
  • Move backwards through time to explore file backups

• Ecological, contextual, broader system structure, e.g.
  • Dropbox: a box you drop everything into
  • Icloud: central mother ship to which everything connects

• Personal relationships, e.g.,
  • Siri as a personal assistant
Example: the desktop metaphor

• Unifying set of concepts employed in graphical user interfaces to help users understand and easily interact with a computer

• Computer monitor → user’s desktop

• Objects → documents, folders
  *you can do things with these objects:*
  • Place documents upon desktop
  • Open documents into a window → paper copy
  • Organize in folders

• Extend desktop with desk accessories → calculator, notepad
Relationship among concepts

• What actions or attributes are shared between objects?
  - E.g. Song, podcast, audiobook all have timelines that users want to navigate (i.e. Fast forward, rewind, etc.)

• Containment and hierarchy
  - E.g., A song is contained by an album
MAPPING of concepts to actual activities

How do the concepts map to what people will actually do?

• One easy way to tell: “run” a task example on it

• Learn:
  • Are these the right objects?
  • Can I do all the operations?
  • Do they match what people want to do?
  • Can I do them in a consistent way?

example: debit machine
TERMINOLOGY

What terms will you use to communicate concepts?

• Terminology should match your concepts

• Choose your terminology and stick to it!
  Easy to go from planning to interface and minimize confusion

• Does your user login to a system with a user-id? A username? A member id? Or an email address?
Conceptual models: interaction and interface

• Interaction type:
  • What the user is doing when interacting with a system.
    • E.G., Command line (how you talk to it), intelligent (function),
      gestural (hardware), touch (both hardware and interaction type)

• Interface type:
  • The kind of interface used to support the mode.
    • E.G. Speech, menu-based, gesture
Interaction types

• Instructing
  Instruct a system and tell it what to do; issuing commands and selecting options (e.g. Print a file, save a file)

• Conversing
  Interacting with a system as if having a conversation (e.g. Search engines, advice-giving systems, help systems, virtual agents)

• Manipulating
  Interacting with objects in a virtual or physical space by manipulating them (e.g. Dragging, selecting, opening, closing and zooming actions on virtual objects)

• Exploring
  Moving through a virtual environment or a physical space (e.g. Google maps, GPS)
INTERFACE TYPES
• Many different kinds (we won’t examine each in detail)
  • Includes: mobile, GUI, touch, tangible, haptic, desktop, command line, data visualizations…

<table>
<thead>
<tr>
<th>Interface type</th>
<th>See also</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Command-based</td>
<td>WIMP and web</td>
</tr>
<tr>
<td>2. WIMP and GUI</td>
<td>Augmented and mixed reality</td>
</tr>
<tr>
<td>3. Multimedia</td>
<td>Multimedia</td>
</tr>
<tr>
<td>4. Virtual reality</td>
<td>Mobile and multimedia</td>
</tr>
<tr>
<td>5. Information visualization</td>
<td>Mobile</td>
</tr>
<tr>
<td>6. Web</td>
<td>Augmented and mixed reality</td>
</tr>
<tr>
<td>7. Consumer electronics and appliances</td>
<td>Shareable, touch</td>
</tr>
<tr>
<td>8. Mobile</td>
<td>Shareable, air-based gesture</td>
</tr>
<tr>
<td>9. Speech</td>
<td>Tangible</td>
</tr>
<tr>
<td>10. Pen</td>
<td>Multimodal</td>
</tr>
<tr>
<td>11. Touch</td>
<td>Speech, pen, touch, gesture, and haptic</td>
</tr>
<tr>
<td>12. Air-based gesture</td>
<td>Touch</td>
</tr>
<tr>
<td>13. Haptic</td>
<td>Virtual reality</td>
</tr>
<tr>
<td>14. Multimodal</td>
<td></td>
</tr>
<tr>
<td>15. Shareable</td>
<td></td>
</tr>
<tr>
<td>16. Tangible</td>
<td></td>
</tr>
<tr>
<td>17. Augmented and mixed reality</td>
<td></td>
</tr>
<tr>
<td>18. Wearable</td>
<td></td>
</tr>
<tr>
<td>19. Robotic</td>
<td></td>
</tr>
<tr>
<td>20. Brain–computer</td>
<td></td>
</tr>
</tbody>
</table>
What does a conceptual model look like?

• However best helps you describe and understand its components:
  • Lists and tables
  • Diagrams
  • Storyboards and sketches
  • Written descriptions
  • Mood boards
  • Physical ‘sketches’

Different methods might capture different parts of more effectively than others

➔ you’ll likely use a combination of more than one!
Conceptual model for a debit machine

- Using a diagrammatic approach
- shows concepts, relationships, terminology
Storyboards and sketching

• Flexible methods for representing conceptual design!
  • Can be used to show what the user is thinking/feeling
  • Communicate metaphors
  • Interface types and styles of interaction
  • Environments and contexts in which system is used
• Can be very low investment
• Note: you don’t need to be good at drawing to communicate your ideas
  • Sketches and storyboards can vary in fidelity
This conceptual design representation emphasizes objects and relationships for an e-ticket system.
Storyboards

Guidelines for storyboards:

- Decide what you want trying to communicate
- Consider *characters, plot, environment, user’s thought process and emotions*
- **Iterate:** start with text and arrows & move up to more involved drawings
Summary:
a good conceptual model:

• Must make sense  
  e.g., *Metaphors that build on something the user knows, and translates well*

• Has to be consistent  
  e.g., *In terminology, in how objects are interacted with.*

• Has a minimal set of concepts  
  *keep it simple as possible; conceptual model will be apparent to user if they can see all of it*

• Focuses on elements of task user wants to do

• *Need to settle on it early in the process*
Activity [20 min]

Check the website for the worksheet

Scenario: imagine: you’ve been hired to (eventually) build a new user web interface for reserving rooms in the UMass CS department.

Task: Create a visual representation(s) of a conceptual model based on your brainstorming
On deck...

• Next class is not going to be a working class, check the website for the updated schedule...
  • Readings (as posted)

• Second project milestone: Ideate
  • due on Oct 15th
Extra slides
System design vs. Interface design?

- System designers and implementers may have more concepts or details going on in the background.
- But conceptual model (and eventually interface) should only contain what users need.
- System concepts should only be included when they can foster a good mental model.
Identifying concepts: object / operation analysis

- Method from Johnson and Henderson
- What are all the ‘concepts’ that a user will need in the system?
- Implication: should be what people use to interact with the interface!

Include: all objects, attributes, operations of tasks that users need to be aware of or understand to use system

- User-understandable entity types (objects, people, …?)
- Attributes of each entity-type
- Operations that users can perform on each type of object
- Note where these concepts may be different for different users

Task examples are a great resource for these!
identifying concepts:
example:

<table>
<thead>
<tr>
<th><strong>objects</strong></th>
<th><strong>attributes</strong></th>
<th><strong>operations</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>songs</td>
<td>album, title, artist, descriptions, currently playing, # times played, date</td>
<td>play, preview, pause, stop, rewind, fast forward, add to play list, send to a</td>
</tr>
<tr>
<td></td>
<td>added to system</td>
<td>friend</td>
</tr>
<tr>
<td>album</td>
<td>title, artist, description, compilation, currently playing, # times played,</td>
<td>play, stop, add to play list, send to a friend</td>
</tr>
<tr>
<td></td>
<td>date added to system</td>
<td></td>
</tr>
<tr>
<td>playlist</td>
<td>title, description, date created, # times played</td>
<td>play, stop, skip song, choose song, send to a friend</td>
</tr>
<tr>
<td>user profile</td>
<td>username, favorite albums, favorite songs, credit card #,</td>
<td>review songs, review albums,</td>
</tr>
</tbody>
</table>
Interaction types

1. Instructing
instruct a system and tell it what to do; issuing commands and selecting options (e.g. tell the time, print a file, save a file)

2. Conversing
interacting with a system as if having a conversation (e.g. search engines, advice-giving systems, help systems, virtual agents)

3. Manipulating
interacting with objects in a virtual or physical space by manipulating them (e.g. dragging, selecting, opening, closing and zooming actions on virtual objects)

4. Exploring
moving through a virtual environment or a physical space (e.g. google maps)
1. Instructing

• Use when:
  • User needs to **tell system** what to do
    *RSP defines as *indirect* (as opposed to ‘direct manipulation’)*

• Common conceptual model:
  • Word processors (open, close, save, etc.)
  • Vcrs/dvd players (play, rewind, pause, etc.)

• Benefit: supports quick and efficient operations
  • Good for repetitive actions on more than one object
  • Must be aware of the possibilities – learned
2. Conversing

• Use when:
  User needs have a dialogue, i.e. Back-and-forth.
  • Really a dialogue, not just a series of options and selections.
  • More of a 2-way conversation than in instructing

• Examples: often implemented with natural language
  • Many online help centers (have you ever been fooled?)
  • Siri (can also be instructing)
  • Edge case: typing queries into a web search engine
  • Compare with: kiosk operation like buying a bus ticket

• Benefit: when/WHY to use?
  • Good for novices, the computer phobic, specialized applications, etc.
3. Manipulating

- **Use when:**
  - Makes sense to *directly manipulate objects*
  - Benefit: leverages what people *do in the real world*; (e.g., Drag/drop)
  - But CAN be used for non-realistic actions too (e.g., Zoom)

- **Principles:**
  - Representation is always available (visible)
  - Incremental, reversible actions (“undo”)
  - Physical actions (drag/drop) rather than syntactic commands

- **Examples of tasks that could use “manipulating”**
  - File operations (open, close, save)
  - Moving selected block of text around on a powerpoint slide
  - Touch interaction with maps (pinch, zoom, slide)
4. Exploring

• Use when:
  • User needs to explore and interact with an ‘environment’.
  • Can exploit user’s previous knowledge of how they move through spaces (digital and physical)

• Examples of tasks that could use “exploring”
  • Finding a location in google maps: 
    using street view
  • Identify location using ‘dot’ on GPS: 
    physically move through actual environment with phone
Interaction types

• Instructing, manipulating most common historically; but conversing and exploring increasingly used

• Not exclusive
  • You can do multiple within one interface for DIFFERENT objects
  • Or for the SAME objects, e.g.,
    ➔ Instructing AND manipulating of files (open, close, save, etc.)
    ➔ Instructing and conversing for help functions
    ➔ Conversing AND exploring for following GPS directions