

Introduction to HCI

Fall 2021

Conceptual Models

Mahmood Jasim

UMass Amherst

mjasim@cs.umass.edu

<https://people.cs.umass.edu/~mjasim/>

© Mahyar with acknowledgements to Joanna McGrenere and Dongwook Yoon

Logistics

- ▶ Presentation slides submission due tonight
- ▶ Presentation format
- ▶ Team responsibilities
 - ▶ Communication
 - ▶ Contribution
 - ▶ Addition to milestone: Write down team members' names and their contribution to the milestone
 - ▶ Representation
 - ▶ Be present during in-class presentations even if you are not presenting
 - ▶ Impact on grading

Learning goals

- ▶ Explain the difference between:
 - ▶ A conceptual design and an interface design
 - ▶ A conceptual model and a user's mental model
- ▶ Given a scenario:
 - ▶ Identify and list the components a conceptual model should include (e.g., metaphors, interaction types, objects/attributes, etc.)

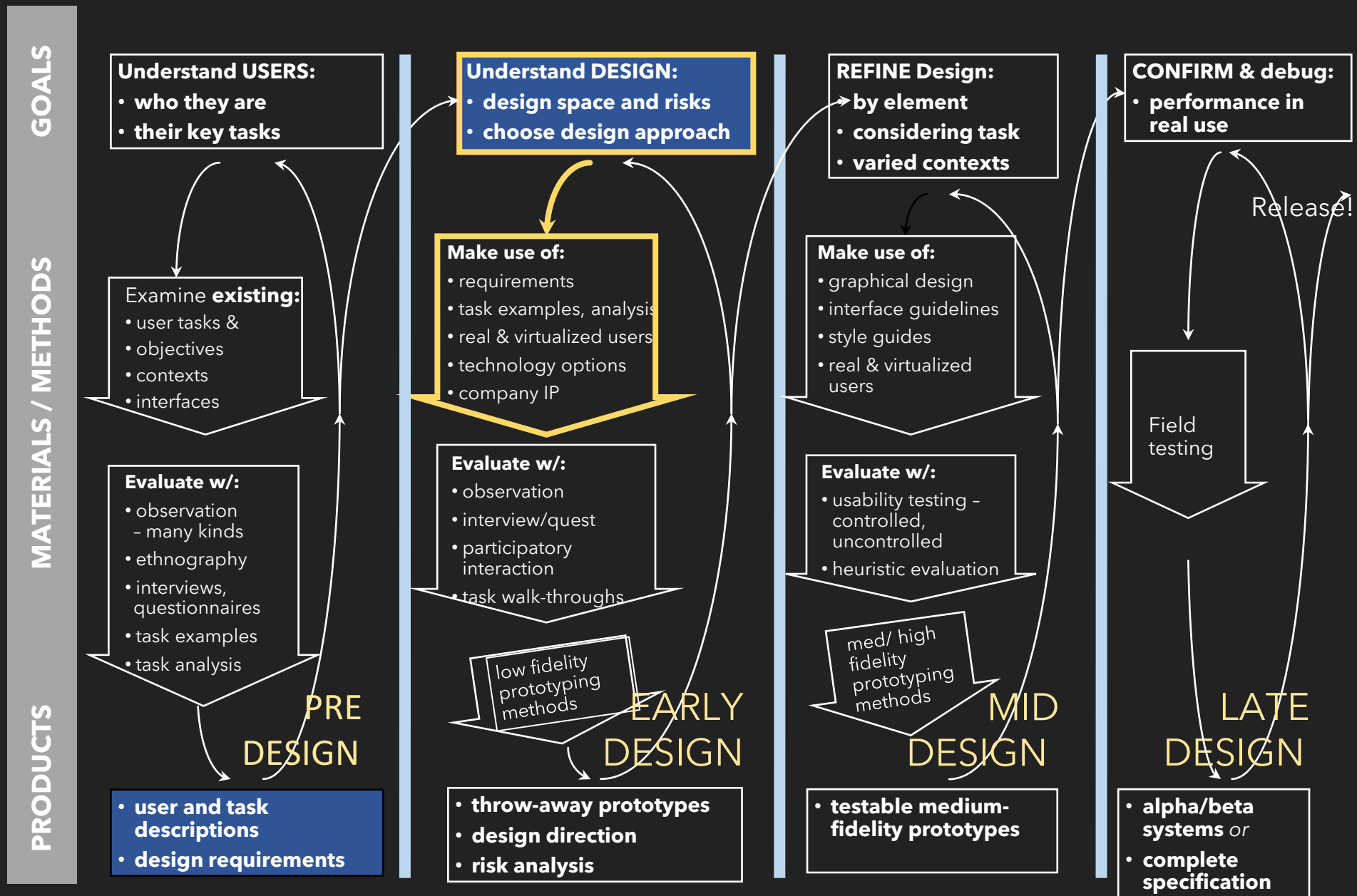
Mental models vs. Conceptual models

- ▶ **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)

Introduction to Conceptual Models

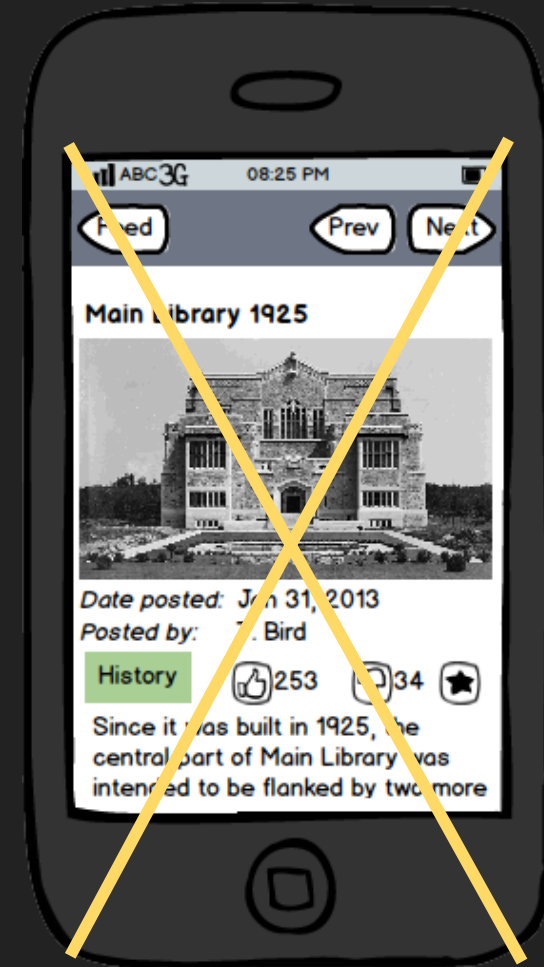


<https://www.youtube.com/watch?v=shSCUNxtn18>



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:

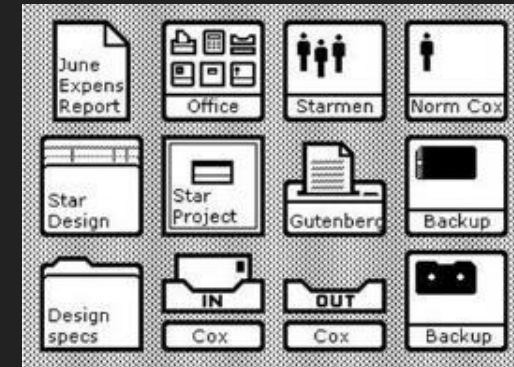
- ▶ Any central design metaphors and analogies; e.g., The “desktop metaphor”
- ▶ Concepts – objects, actions you can do to them; user roles; attributes of both; e.g., Files and folders; both can be opened, have names
- ▶ Relationships among concepts; e.g., Files are contained in folders
- ▶ Mappings 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
- ▶ **Interactions**
 - ▶ Swipe to turn page in an ebook
 - ▶ Move backwards through time to explore file history
- ▶ **Ecological, contextual, broader system structure**
 - ▶ Dropbox: a box you drop everything into
 - ▶ icloud: central mother ship to which everything connects
- ▶ **Personal relationships**
 - ▶ 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



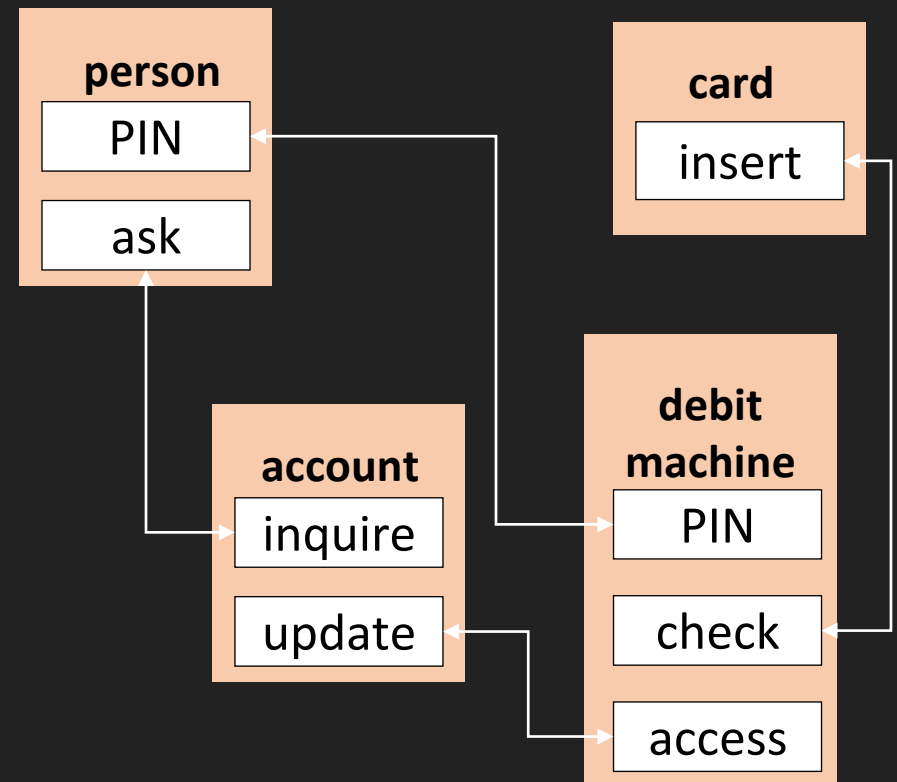
Relationship among concepts

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

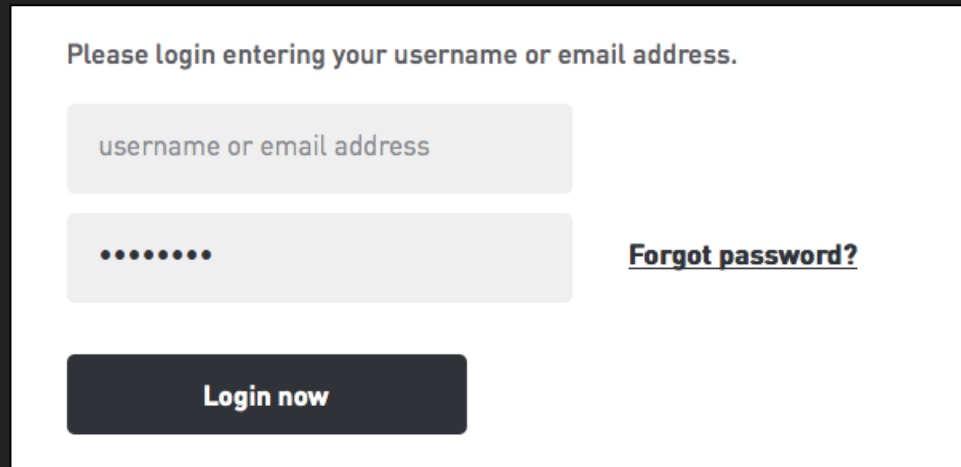


example: debit machine

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?



Please login entering your username or email address.

username or email address

..... [Forgot password?](#)

Login now

Conceptual models: interaction and interface

- ▶ Interaction type:
 - ▶ What the user is doing when interacting with a system
 - ▶ 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.
 - ▶ 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...

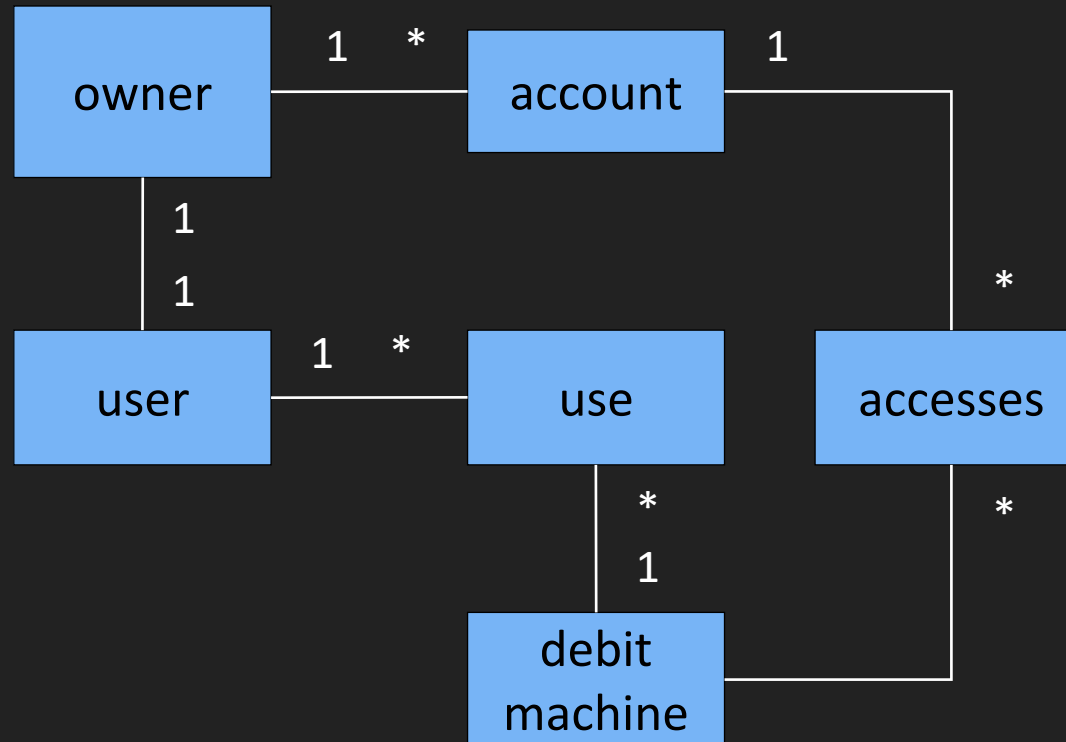
Interface type	See also
1. Command-based	
2. WIMP and GUI	
3. Multimedia	WIMP and web
4. Virtual reality	Augmented and mixed reality
5. Information visualization	Multimedia
6. Web	Mobile and multimedia
7. Consumer electronics and appliances	Mobile
8. Mobile	Augmented and mixed reality
9. Speech	
10. Pen	Shareable, touch
11. Touch	Shareable, air-based gesture
12. Air-based gesture	Tangible
13. Haptic	Multimodal
14. Multimodal	Speech, pen, touch, gesture, and haptic
15. Shareable	Touch
16. Tangible	
17. Augmented and mixed reality	Virtual reality
18. Wearable	
19. Robotic	
20. Brain-computer	

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!

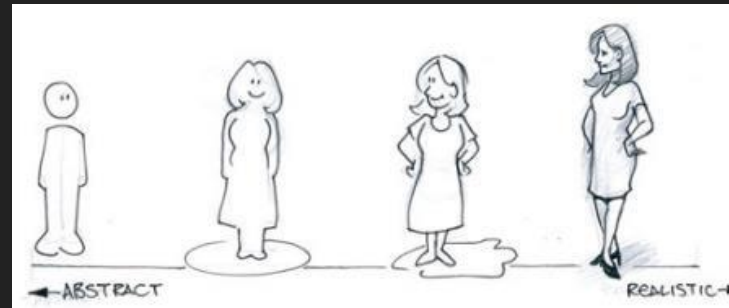
Conceptual model for a debit machine

- ▶ Using a diagrammatic approach
- ▶ Shows concepts, relationships, terminology

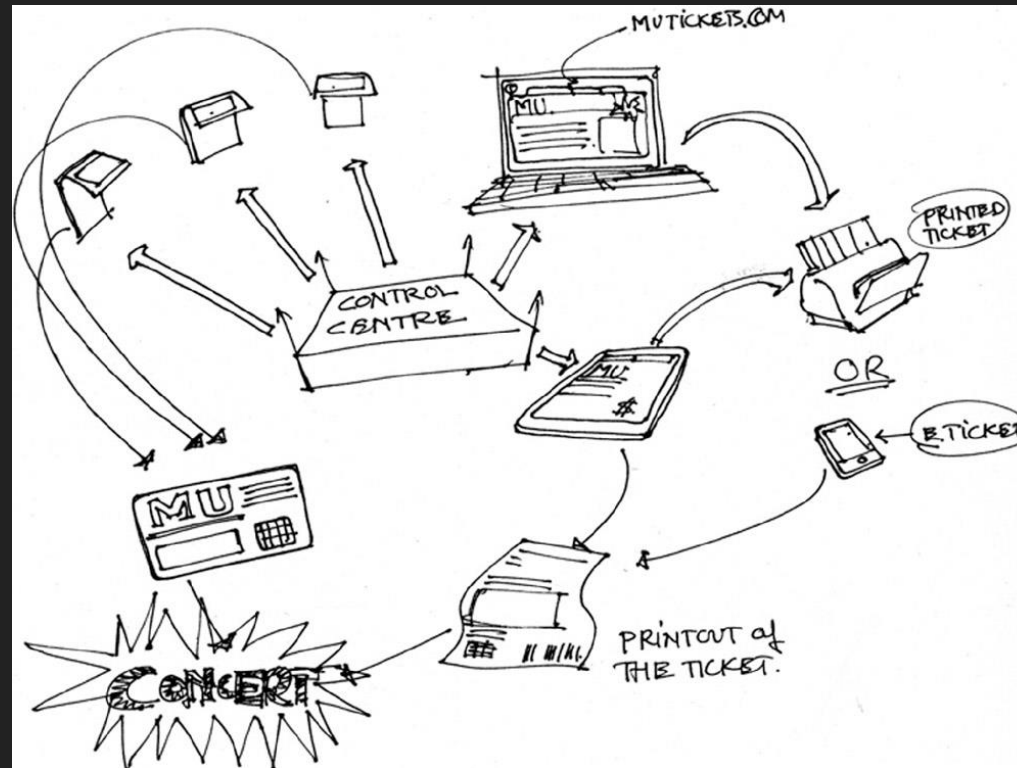


Storyboards and sketching

- ▶ 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



Akshay Sharma, Virginia Tech Department of Industrial Design from The UX Book: Process and Guidelines for Ensuring a Quality User Experience, Rex Hartson and Pardha S. Pyla

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



Ben Crothers. Storyboarding & UX -Part 2: Creating your own -

<http://johnnyholland.org/2011/10/storyboarding-ux-part-2-creating-your-own/>

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***

In-Class activity

- ▶ Imagine you've been hired to build a new web-based user interface for reserving rooms in the UMass CS department
- ▶ In this system, users must be able to:
 - ▶ log on with their department ID
 - ▶ see what rooms exist (list or map view)
 - ▶ see and search room availability
 - ▶ reserve a room (if it is available), and receive an email confirmation sent to their department ID
 - ▶ hold one future room reservation at a time
 - ▶ see their own future reservation, if any
 - ▶ If a room has already been reserved by someone else, students should not be able to find out who has reserved it, but users with tech staff credentials should be able to find this information

In-Class activity

- ▶ Work in groups
- ▶ What could a conceptual model for a room-booking system contain?
 - ▶ brainstorm on each dimension of conceptual models that we've discussed
- ▶ Create a visual representation(s) of a conceptual model based on your brainstorming
- ▶ Link to worksheet - <https://tinyurl.com/f2jek4uf>

Optional Reading

- ▶ Conceptual Models: Begin by designing what to design, Johnson and Henderson
 - ▶ https://drive.google.com/file/d/1XPvLAlayfhW56BTd6CxGN_wwY9s4EVhW/view?usp=sha
- ▶ How UX storyboards can transform your creative process
 - ▶ <https://uxdesign.cc/ux-storyboarding-2ce43875f3de>

Additional Information

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:

objects	attributes	operations
songs	album, title, artist, descriptions, currently playing, # times played, date added to system	play, preview, pause, stop, rewind, fast forward, add to play list, send to a friend
album	title, artist, description, compilation, currently playing, # times played, date added to system	play, stop, add to play list, send to a friend
playlist	title, description, date created, # times played	play, stop, skip song, choose song, send to a friend
user profile	username, favorite albums, favorite songs, credit card #,	review songs, review albums,

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