Conceptual Models & Conceptual Design

CPSC 544 Fundamentals in Designing Interactive Computation Technology for People

Coming up

This week (Oct 17)

- **Sun 10/15**:
  - Team deliverable: Tasks, Requirements and Personas report
  - Researcher Journal #8

- **Wed 10/18**
  - Draft conceptual models for team walkthroughs (in-class activity)

Coming Up

- **Sun 10/22**:
  - Researcher Journal #9

- **Tues 10/24**
  - Researcher Journal #10
  - Team deliverable: Finalized Conceptual Model & Sketches

Big Ideas

A conceptual model is a high-level description of how a system is organized and operates.

- Johnson & Henderson, 2002, p. 26

- 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.
- **Interface design translates the CM** into things we can see and interact with.

Why conceptual design:

If the designers take the trouble to design and refine a conceptual model for the system ***before*** *they design a user interface for it,* ***users will be able to more quickly "figure it out."***

Furthermore, the model they "figure out" will be ***more like the one the designers intended****.*

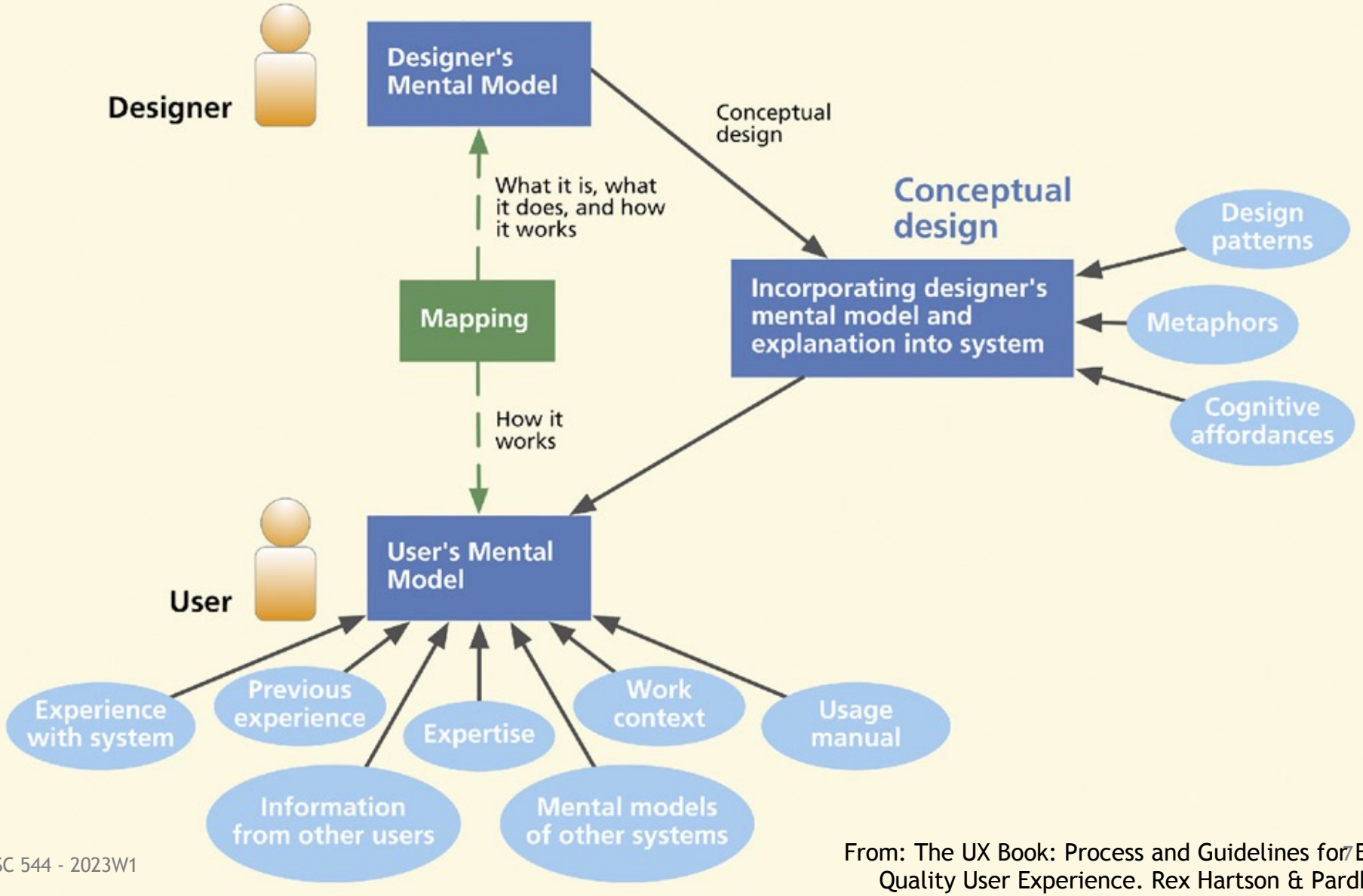
- Johnson & Henderson, 2002, p. 26

Learning Goals

- explain the purpose of a **conceptual model** and how it differs from **a user's mental model.**
- **explain the difference** between a conceptual model and an interface design.
- what are the risks and limitations of **getting conceptual design wrong**?
- **Know components** a conceptual model should include
  metaphors, interaction types, objects/attributes, etc;
  given a scenario, identify examples of each.
- [extra] be able to perform an **object/operation analysis**
- give examples of methods you could use to represent a conceptual model.

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 MM***
  - 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)



From: The UX Book: Process and Guidelines for Ensuring a Quality User Experience. Rex Hartson & Pardha S. Pyla

What is conceptual design?

- Crossing the gap from **requirements** to a **solution**
- Starts with brainstorming; multiple iterations to narrow down
- A **conceptual model** is an outcome of conceptual design
  - sometimes people will call this a ‘conceptual design’
- A conceptual model
  - can take many different forms
  - be built through many approaches
  - is essentially a **set of ideas**

A conceptual model can include:

- Any central design **metaphors** and analogies
  e.g. the “desktop metaphor”
  johnson & henderson
- **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 users interact with it?
  e.g. give commands, perform operations, explore
- **Interface** types; is it/should it be constrained? How would different interfaces affect result?

A conceptual model **excludes**

- low level presentation
- implementation details
- menu and screen designs
- widgets
- etc.

if you start here,
you will get into trouble

How does the CM impact the interface design?
*The CM is the interface's **bones***

- Creates a foundation for what needs to be in interface:

  "If it isn't in the conceptual model, the system should not require users to be aware of it."

  - Rare for a conceptual model to be completely new
    - common ways to do things will inform your model AND design
    - e.g., 'shopping carts'
  - **The conceptual model** will impact the architecture of the system: the eventual interface designs possible.
  - But you can always implement the SAME conceptual model
    With different looks and feels, different widgets, different layouts

ACTIVITY: Discovering an interface design

what conceptual design
*did the designer's **start with?***

activity

StickIT: Mockup for a 'ubiquitous notes' mobile app.

- users post 'sticky' notes around UBC
- other users pick up notes; interact with notes

StickIT. CPSC 544 project,
shared courtesy of Dawson, Link & van Rossum, 2014

System design vs. Interface design?

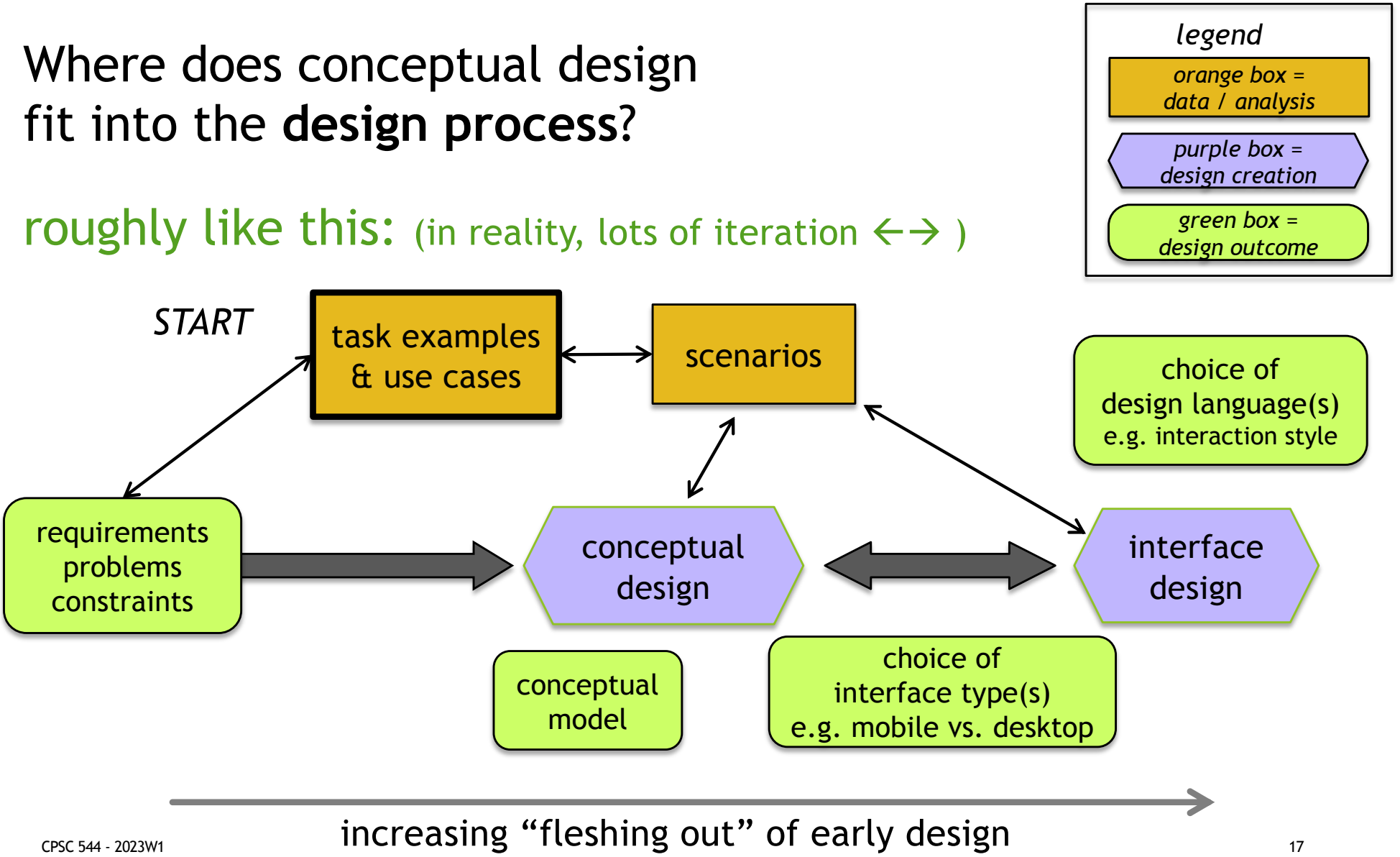
- System designers and implementers may have lots of concepts or details, going on in the background
- But conceptual model (and eventually interface) **should only contain what users need**
- System concepts should only be included when they can foster a **good mental model**

For a CM to work, it must…

- Make sense
  e.g., metaphors that build on something the user knows, and translates well
- Focus on elements of task user wants to do
- Be consistent
  e.g., in terminology, in how objects are interacted with, etc.
- Have a minimal set of concepts
  keep it simple as possible;
- Get locked down EARLY in the process

Where does conceptual design fit into the **design process**?

roughly like this: (in reality, lots of iteration ←→)



conceptual model components: **Objects** and **Actions**

Identifying concepts: object / operation analysis

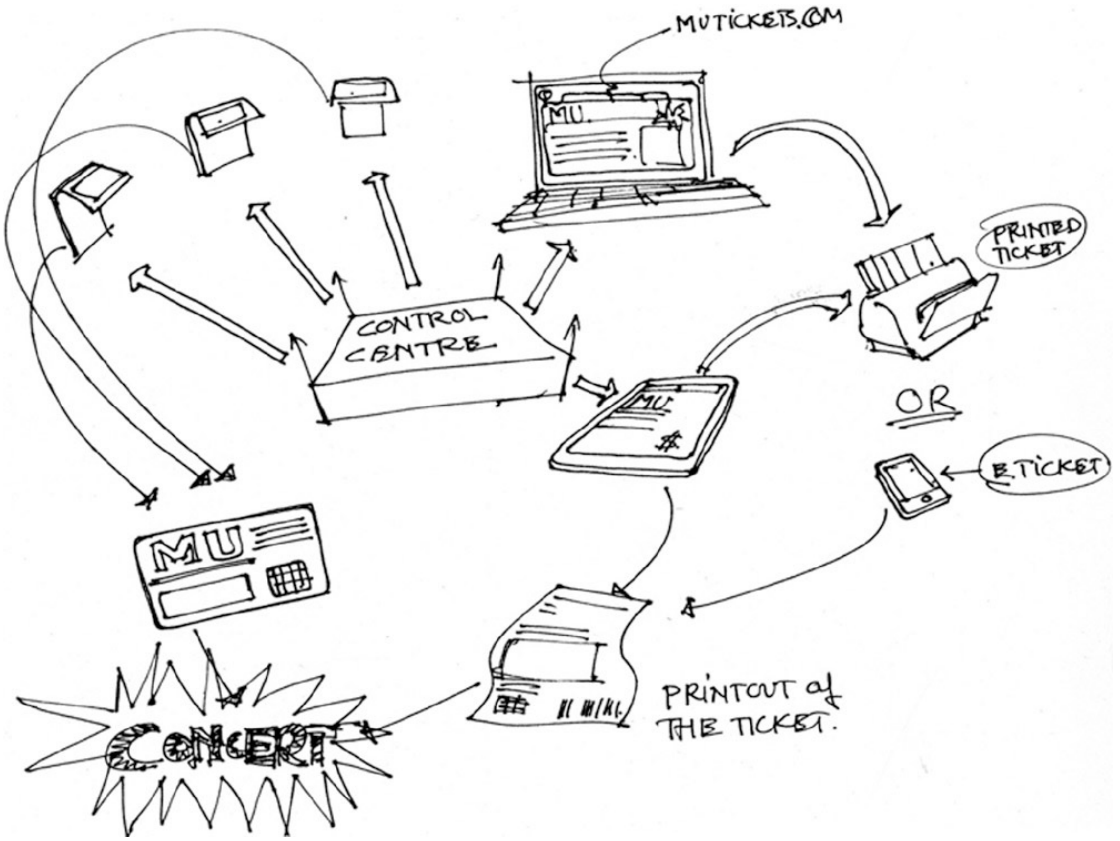
- Method from Johnson & Henderson
  - ➔ What are all the 'concepts' that a user will need in the system?
  - ➔ Implication: be what people use to interact with the interface!

- 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 …

- Some of these concepts different by user
- There may be multiple ways to organize this (multiple CM's)!

Example: Objects and relationships for an e-ticket system



Hartson, R., & Pyla, P. (2012). *The UX Book: Process and guidelines for ensuring a quality user experience*. Elsevier. Chapter: Process and Guidelines for Ensuring a Quality User Experience, Akshay Sharma, p. 288

Work through example as class:

personal digital music streaming system

what concepts does the user need to know / see?

objects	attributes	operations
songs	album, title, artist, descriptions, currently playing, # times played, date added to system, …	play, preview, pause, stop, backup, fast forward, add to play list, send to a friend

More about conceptual model components

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 users 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

Familiar concepts you can rely on to help users understand and interact with the system

Many kinds:

- 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

Seems pretty obvious now!

- 1970: idea from Alan Kay @ Xerox PARC
- Next 10 years: innovated at PARC
- 1981: **Xerox Star** = 1st commercial system

- At that time, with a computer you could:
  - edit a text document, and print it
  - do simple graphics ("paint")
  - compute using something like a spreadsheet
  - mouse for input (not just a keyboard); windowing systems

- But the 'personal computer' didn't exist.

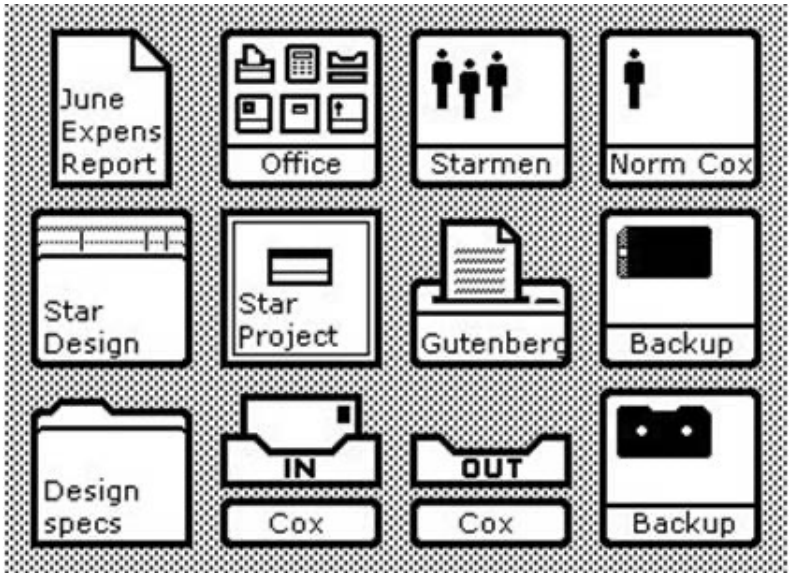
  Most people had never used a computer.

  If they had, it was a "command line" interface, usually shared (mainframe).

  → **Unify** these operations into something **comprehensible.**

This metaphor
made it clear to users what they could do with the system

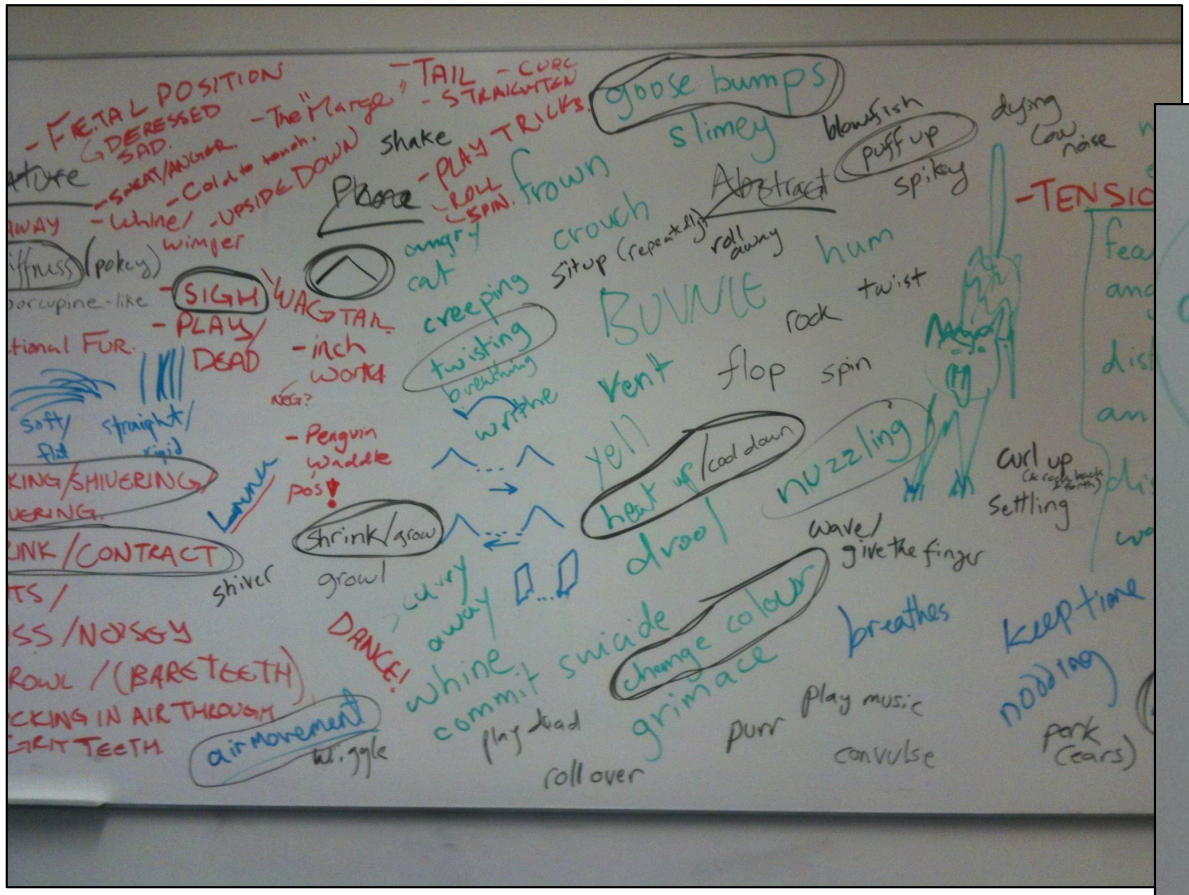
Desktop



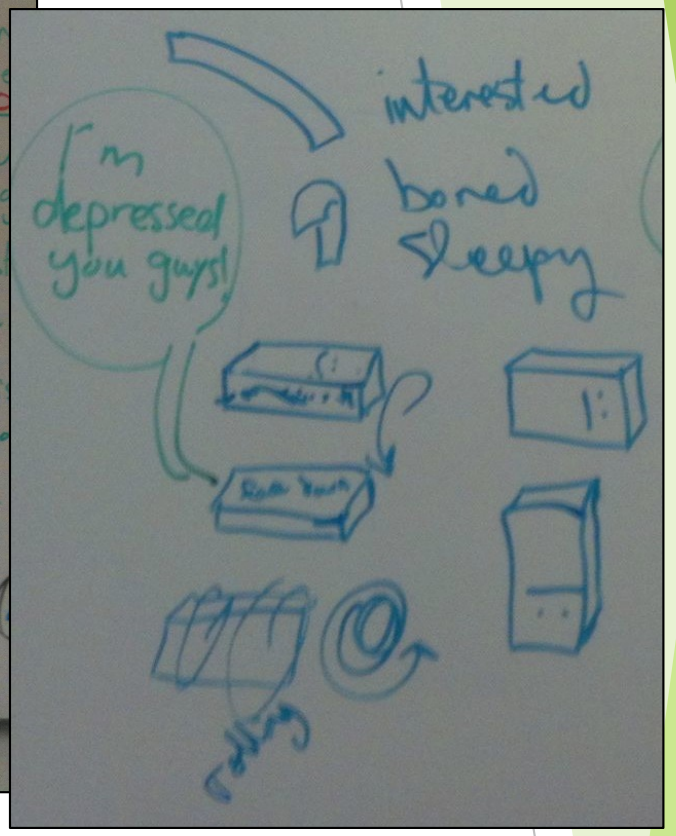
- **icons** graphically represent:
  - **files** – click to open
  - **places** to put them:
    folders, in/outbox, printer
  - **people and groups**:
    email, collaboration
- *how*: **direct manipulation** –
  same as on a real desk; enabled by **mouse**

Another new invention

Metaphors for expressing emotion in a mobile phone



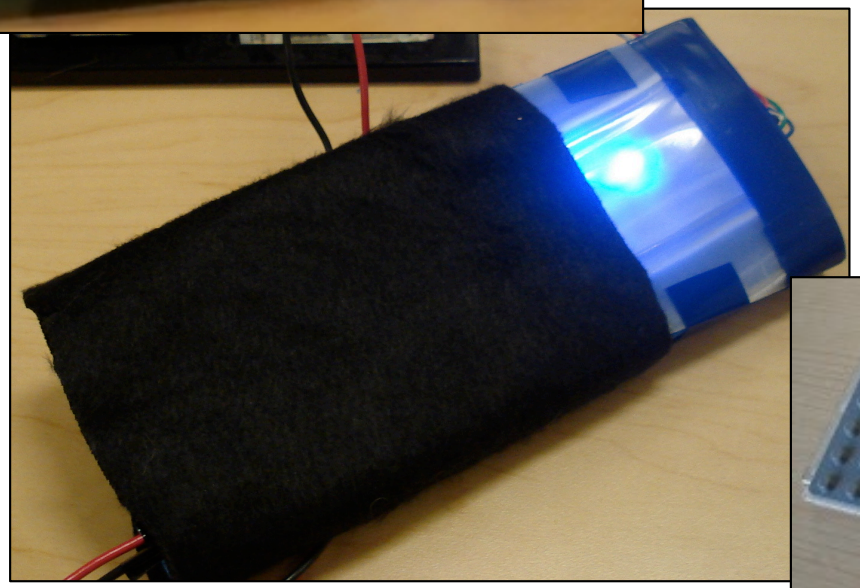
started as a list of words

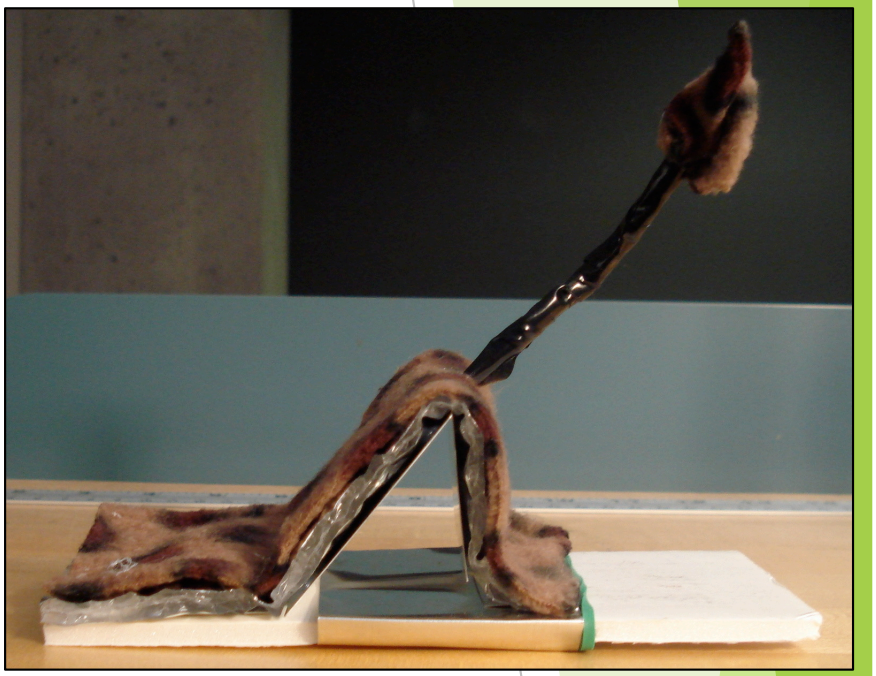


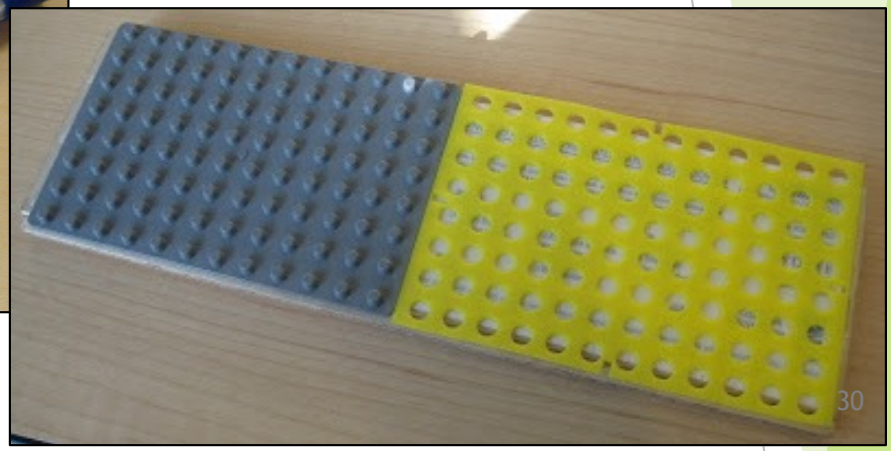
… later became sketches

Dawson, J.Q., Ferstay, J., Link, J., Haddad, S.
Early brainstorming for the DEEVA project (CPSC 543)

Metaphors *were finally expressed in physical form*







Dawson, J.Q., Ferstay, J., Link, J., Haddad, S.
Early concepts for the DEEVA project (CPSC 543)

RELATIONSHIPS 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.)
  - example interface design implication: make the interaction work the same for all, so user recognizes it.
- containment and hierarchy
  - e.g., a song is **contained** by an album
  - interface design implication: represent this containment in the actual design of the interface

Relationships cont. . .

- how do objects, attributes, actions, etc. vary in importance?
  - e.g., in managing a music player, playing songs is a frequent task, while reviewing albums may be infrequent
  - possible interface design implication: frequent tasks and actions should be easy to access; less frequent can be hidden if space is limited

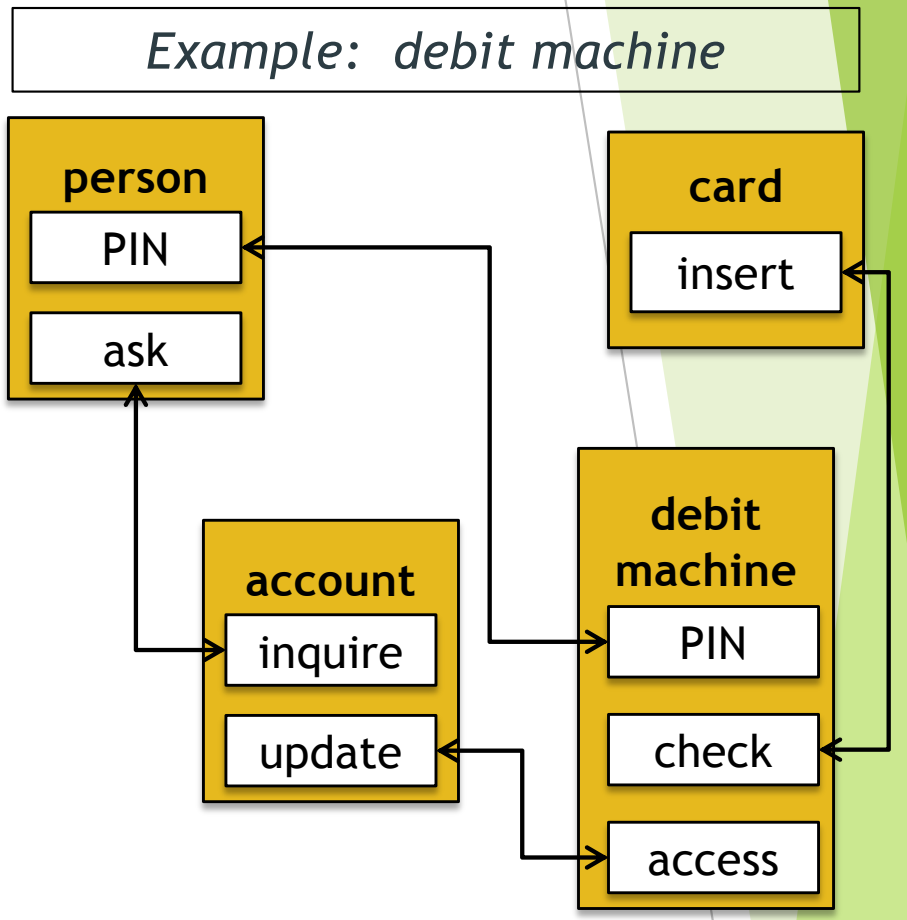
MAPPING of concepts to actual activities

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

"Run" a task example on it

In order to learn:

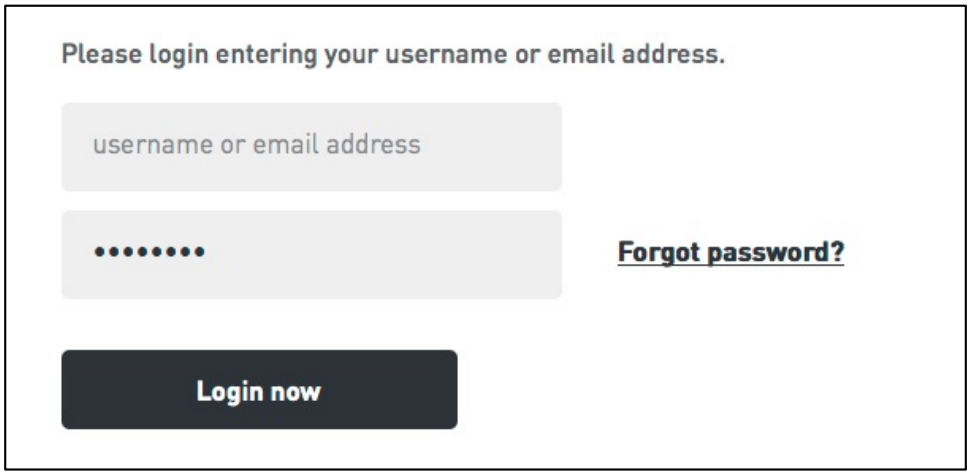
- Are these the right **objects**?
  Do they match what people actually have?
- Can I do all the **operations**?
  Do they match what people want to do?
- Can I do them in a **consistent** way?
- What will the **experience** be like?
  In what **context** will it take place?



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?

What do Conceptual Models look like?

What does a conceptual model look like?

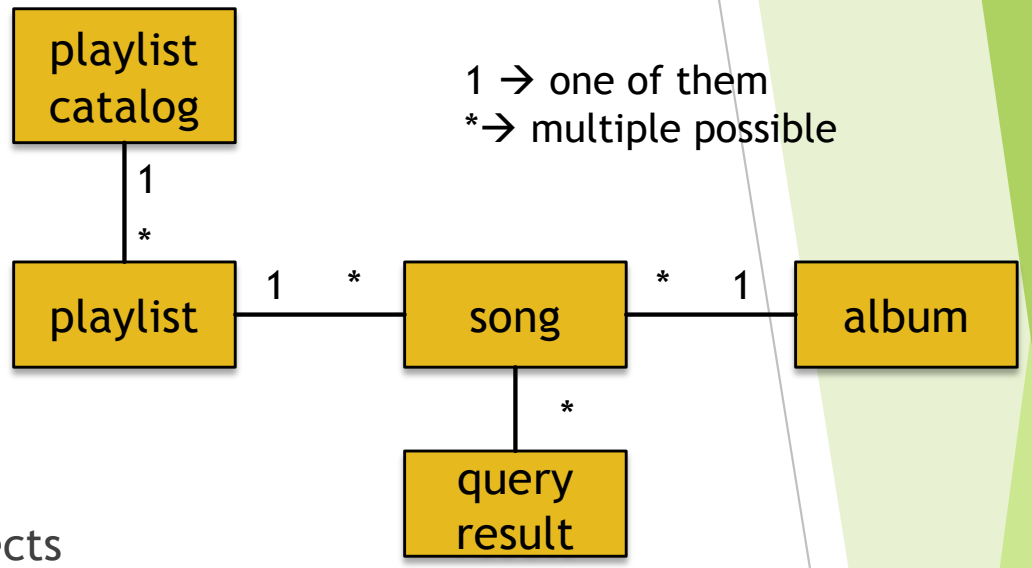
- However best helps you describe and understand its components:
  - lists and tables
  - **diagrams**
  - storyboards and sketches
  - written descriptions
  - mood boards
  - physical ‘sketches’

Different methods might capture different parts of more effectively than others

➔ Valuable to use (or at least try) more than one

concepts, relations, terminology

one possible conceptual model representation for a music player:



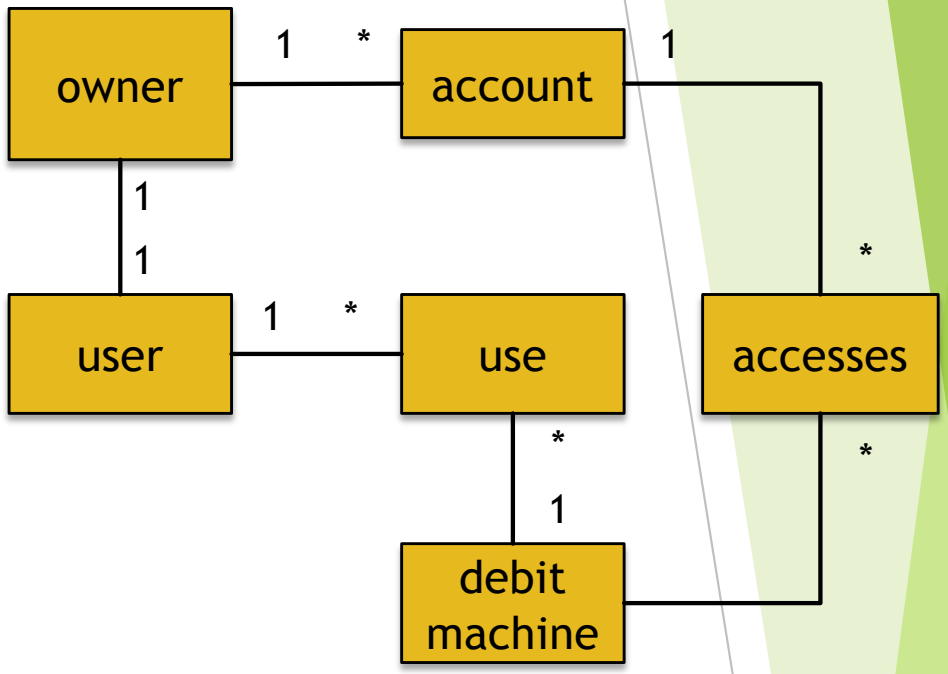
- **objects** represented by boxes
- lines and labels indicate **relations**
- **terminology** = names given to objects

- This CM diagram does *NOT* show what you can do with it;
- It describes what system **consists of** and how it is **organized.**

CM for a debit machine

Another CM:

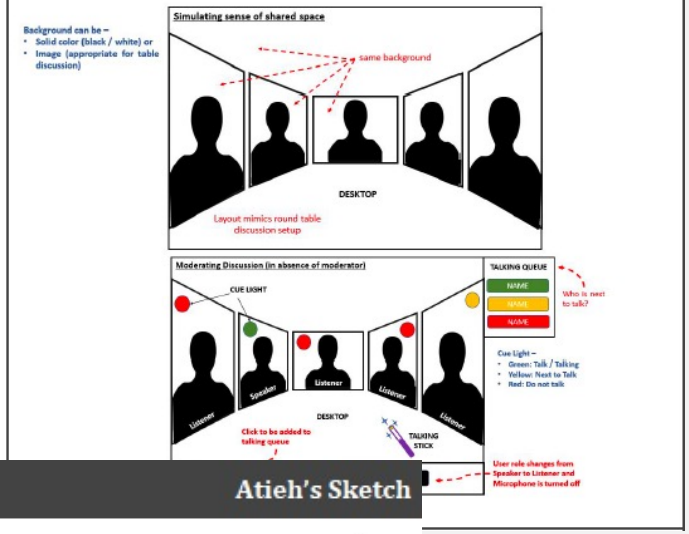
- Uses a diagrammatic approach
- Shows concepts, relationships, terminology



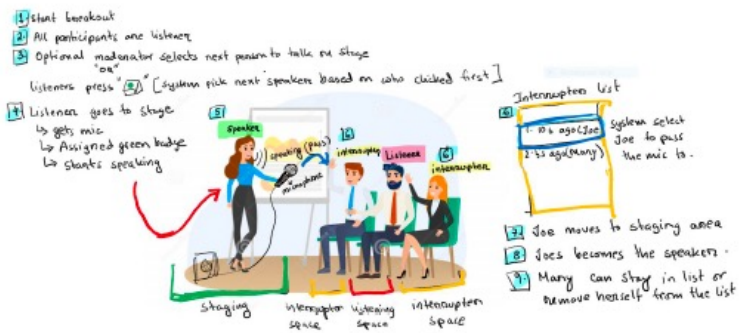
CM Example from 544: Team Pentagon

Many approaches used

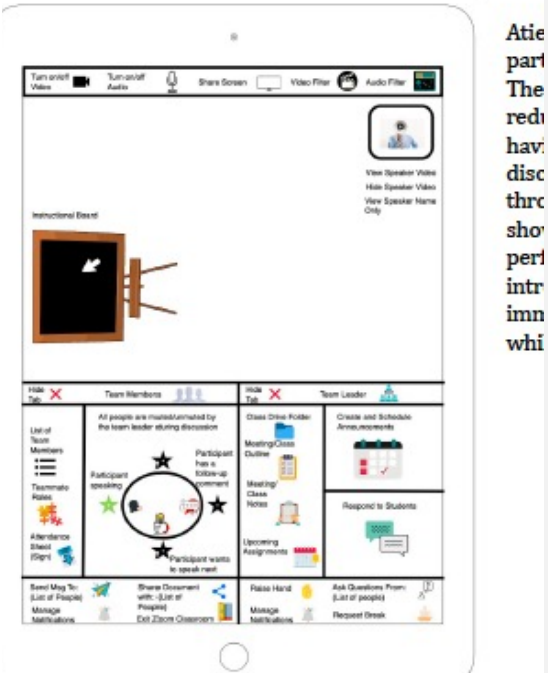
Anika's Sketch



Syed's Sketch

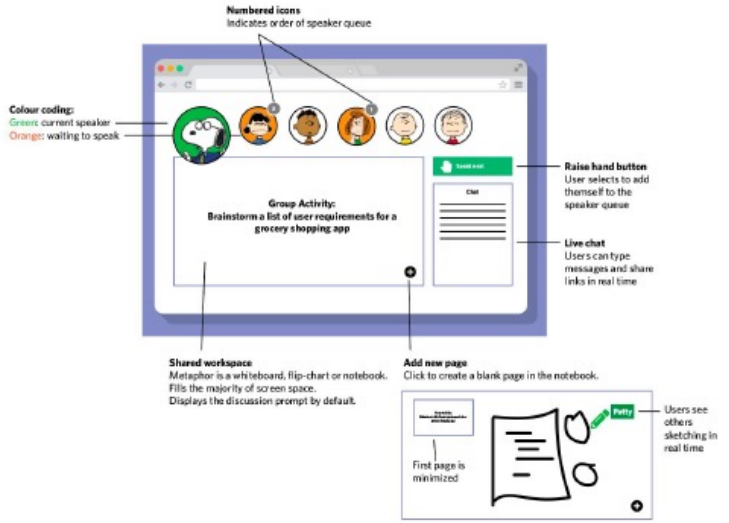


Atieh's Sketch

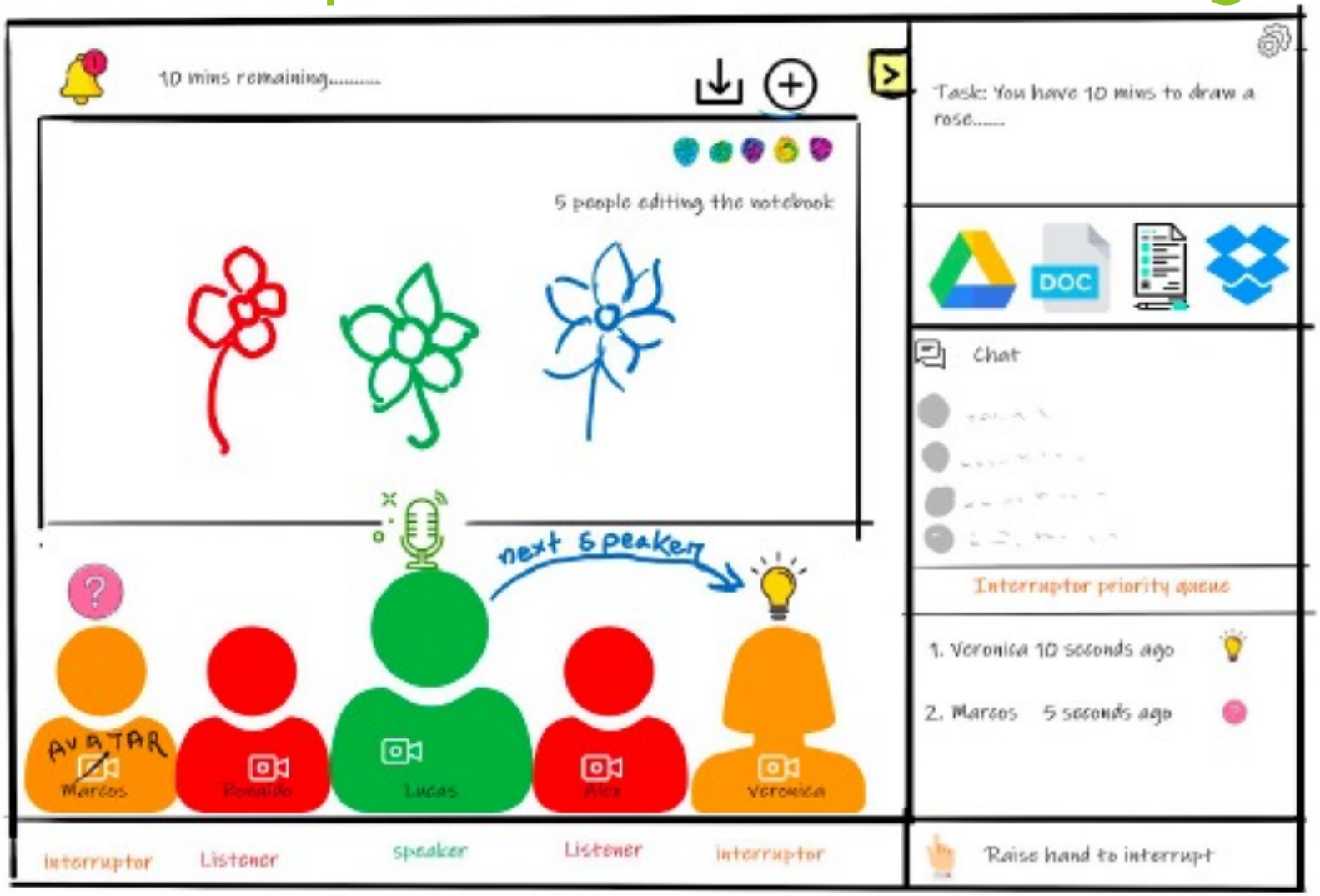


Oloff's Sketch

Jessica's Sketch



Final conceptual model - Team Pentagon



ACTIVITY: BUILDING a conceptual model

Reserving student study rooms

1. Brainstorm: what **concepts** might a conceptual model for this system contain?

2. Create **a visual representation(s)** of a conceptual model based on your brainstorming.
   - For example, a STORYBOARD or OBJECT+RELATIONSHIP diagram.

3. Define a simple **task example** (a story of someone carrying out this activity), then "run" it on your CM. Is your CM up to it?

See handout for more background