

# introducing concepts

Daniel Jackson · Autodesk Oslo Workshop · August 25-26, 2025

a UX puzzle  
Backblaze

# Backing up on Backblaze

Backblaze

dnj@mit.edu 

!



You are backed up as of: 5/17/23, 4:26 PM

Please Wait

Restore Options...

Settings...

Selected for Backup: 916,605 files / 211,505 MB

Backup Schedule: Continuously

Remaining Files: 916,605 files / 211,505 MB

Version History: 30 days [Upgrade](#)

Manage account at [Backblaze.com](#)

Questions? [Help Center](#)

Your data is NOT backed up.

Buy [Already bought?](#) ?



&lt; &gt;



## Backblaze Backup



Search

was  
modification  
at 10pm saved?



You are backed up as of: 6/6/22, 10:10 PM  
Currently backing up newer files

dnj@mit.edu

is backup  
running or not?

Pause Backup

Restore Options...



Selected for Backup: 509,021 files / 2,379,995 MB

Backup Schedule: Continuously

Remaining Files: 0 files / 0 KB

Transferring: photo.0259-22.RAI

huh?

Settings...

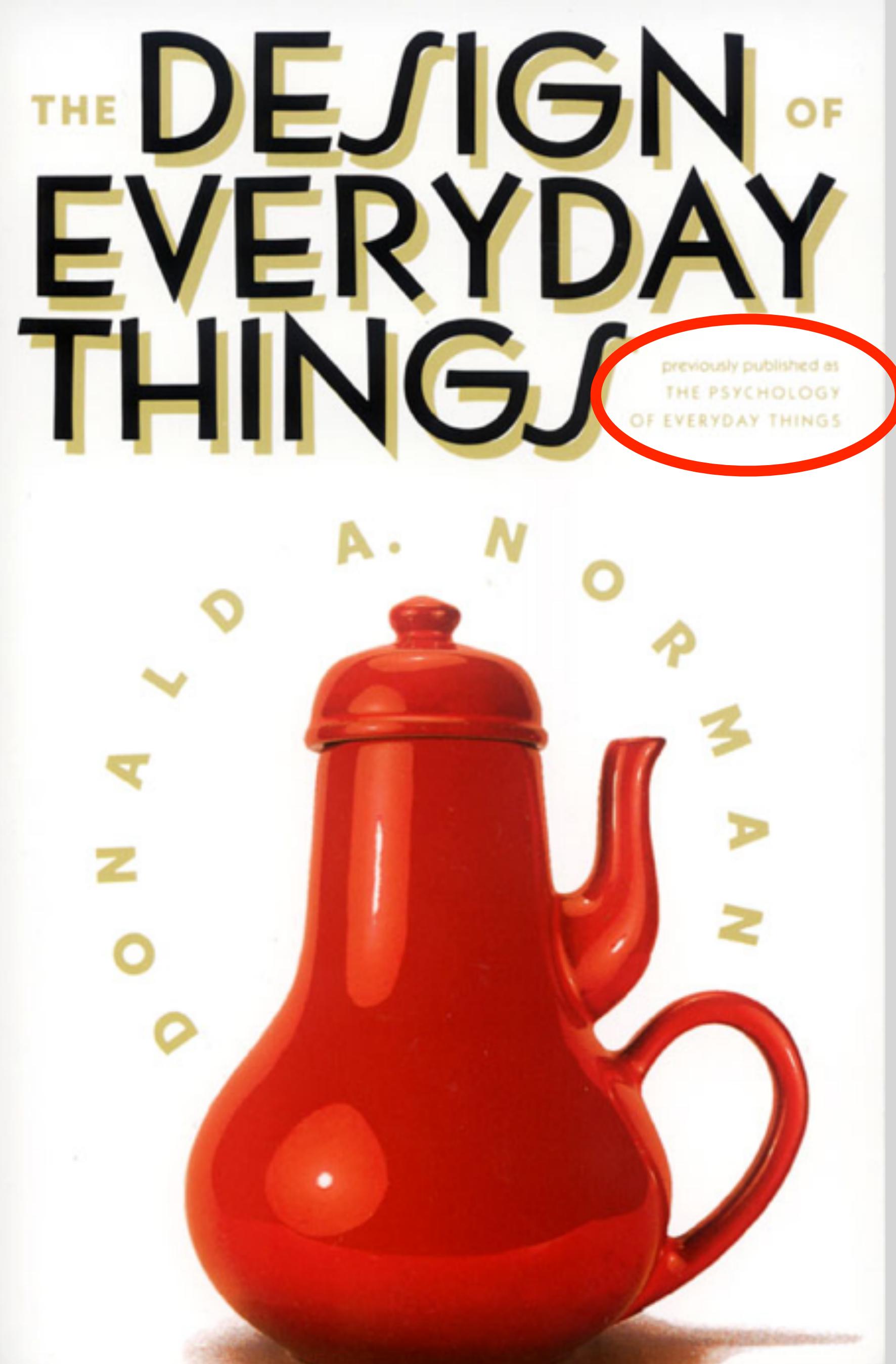
What is being backed up?

How long will my first backup take?

View files and manage account at: [Backblaze.com](https://www.backblaze.com)



conceptual models  
solving Backblaze



When the designers fail to provide a conceptual model, we will be forced to make up our own, and the ones we make up are apt to be wrong.

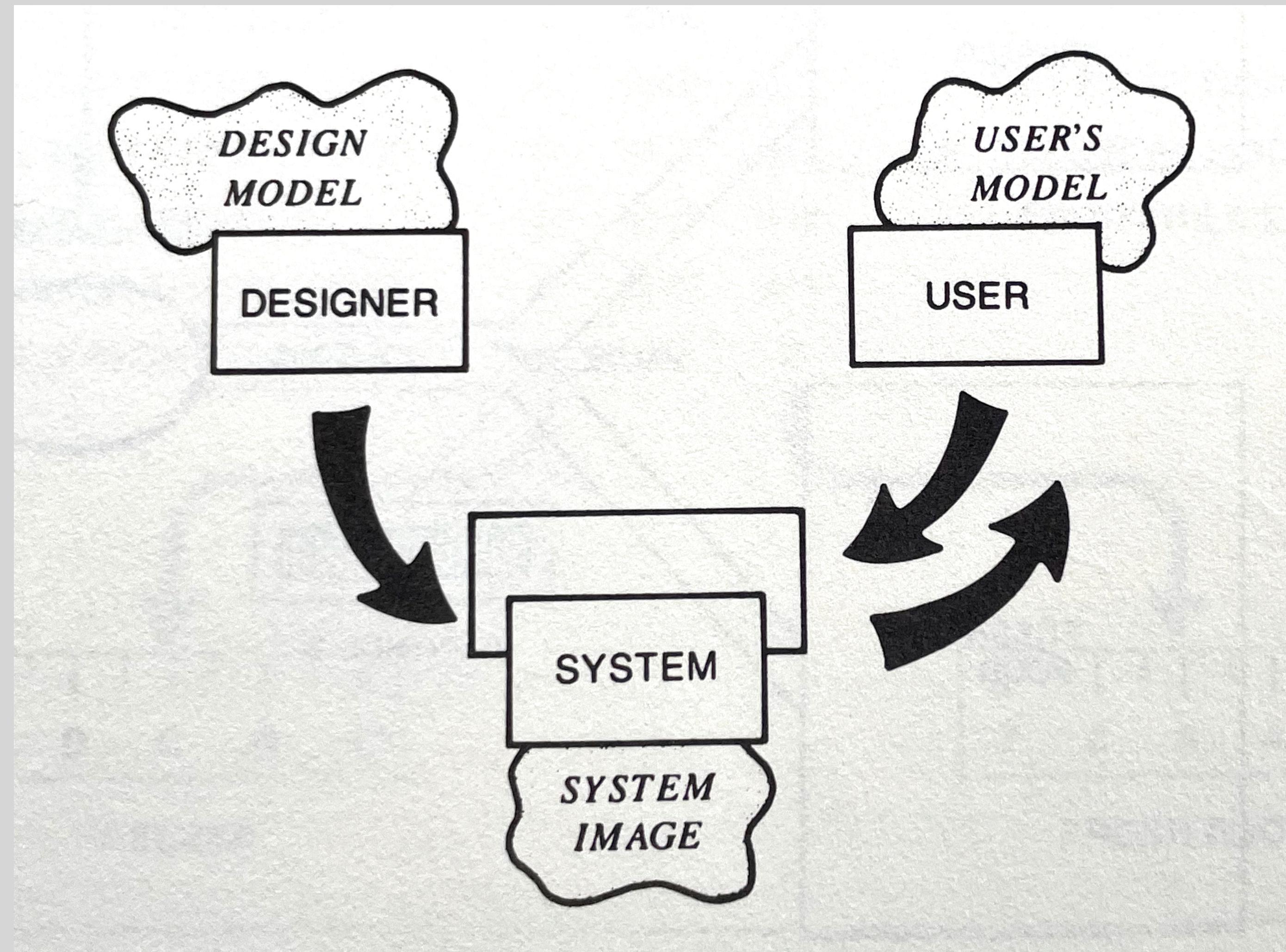
**Conceptual models are critical to good design.**

preface to 2013 edition



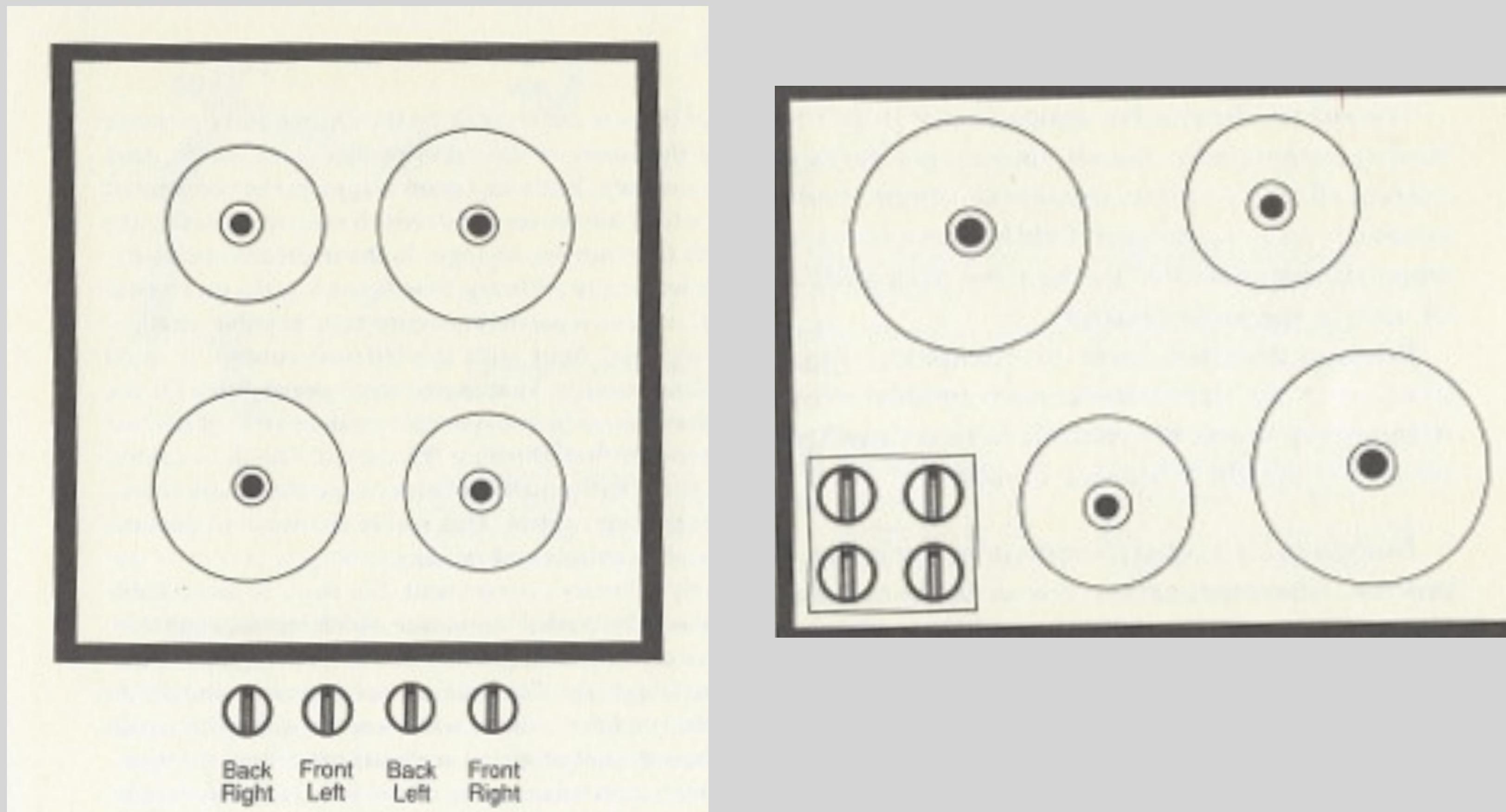
Donald Norman

# the “system image”



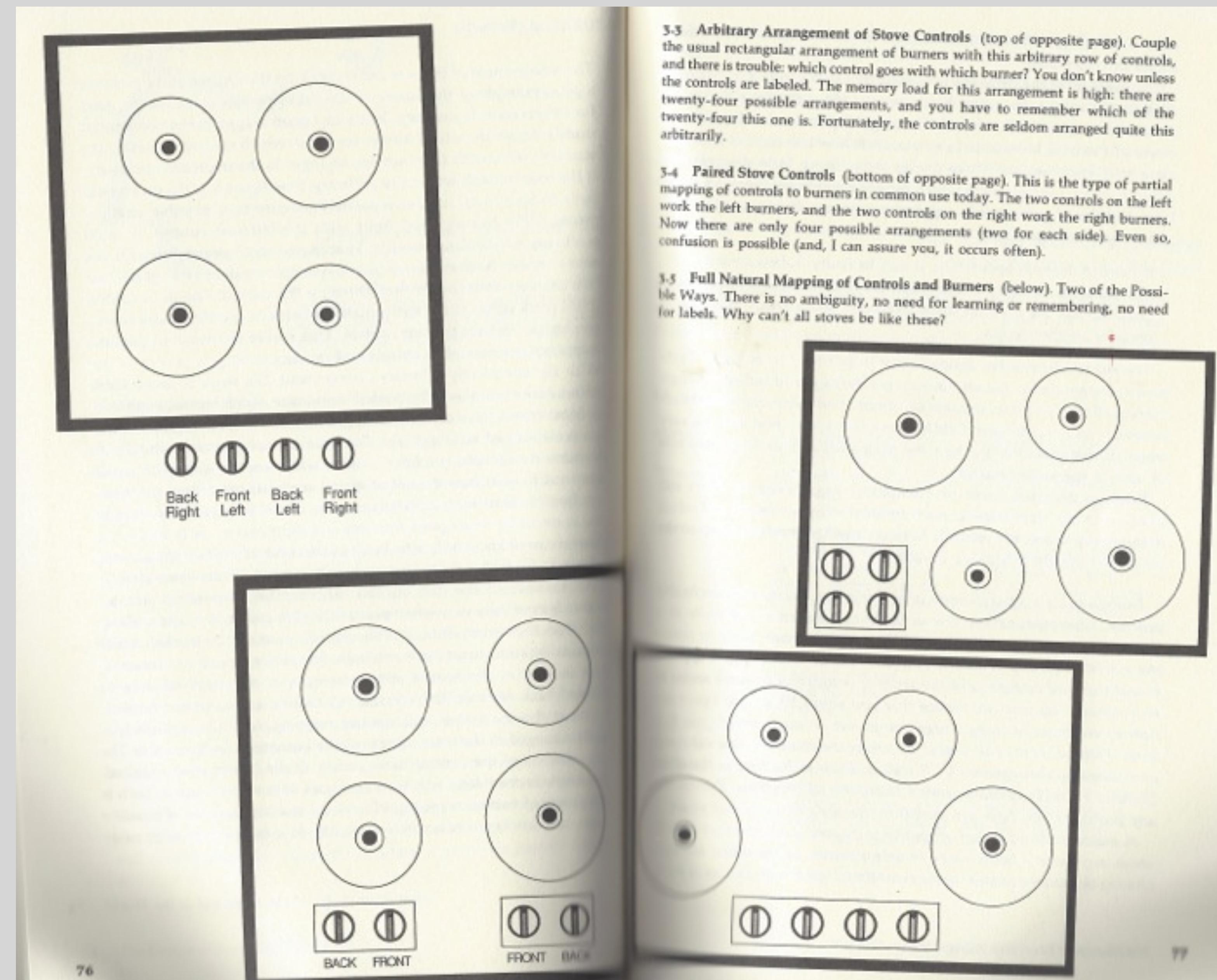
from *The Design of Everyday Things* (1988)

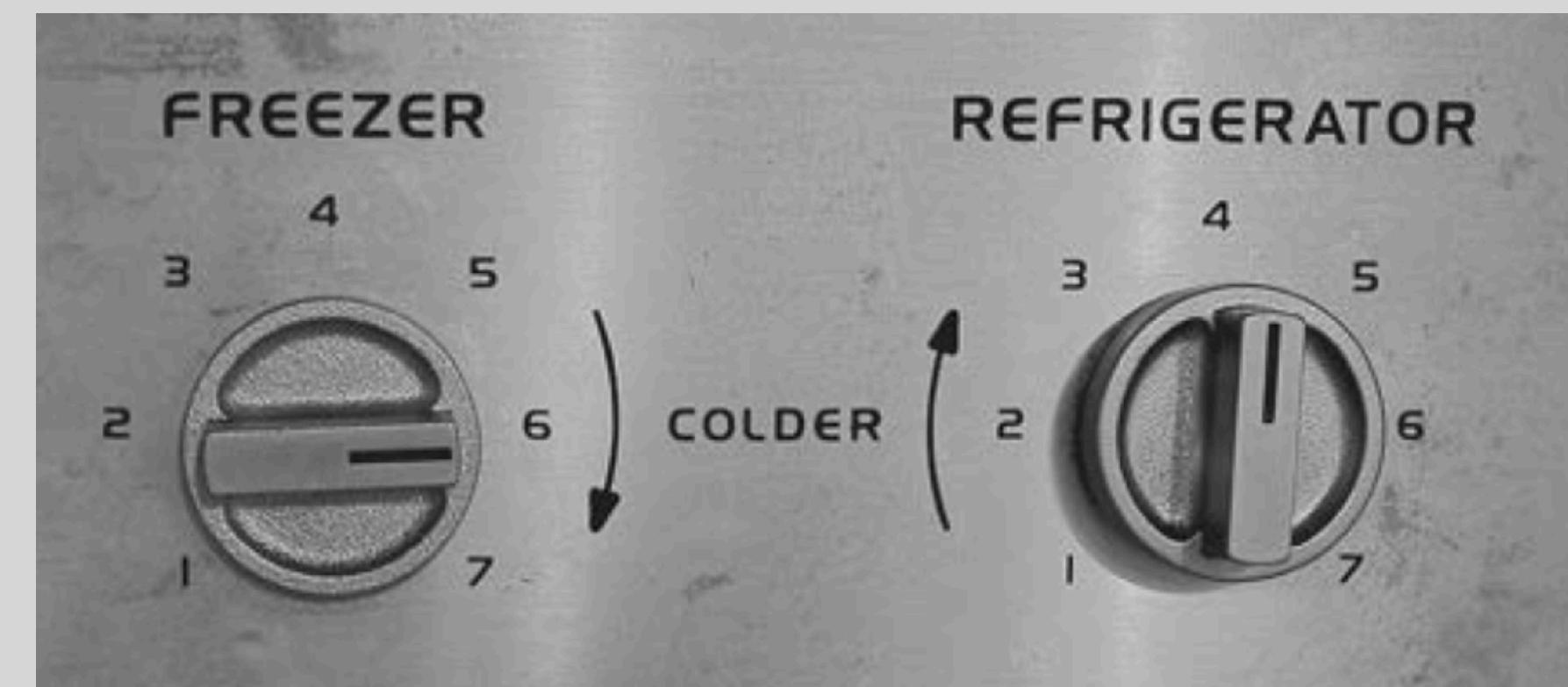
# mapping: one strategy to improve the system image



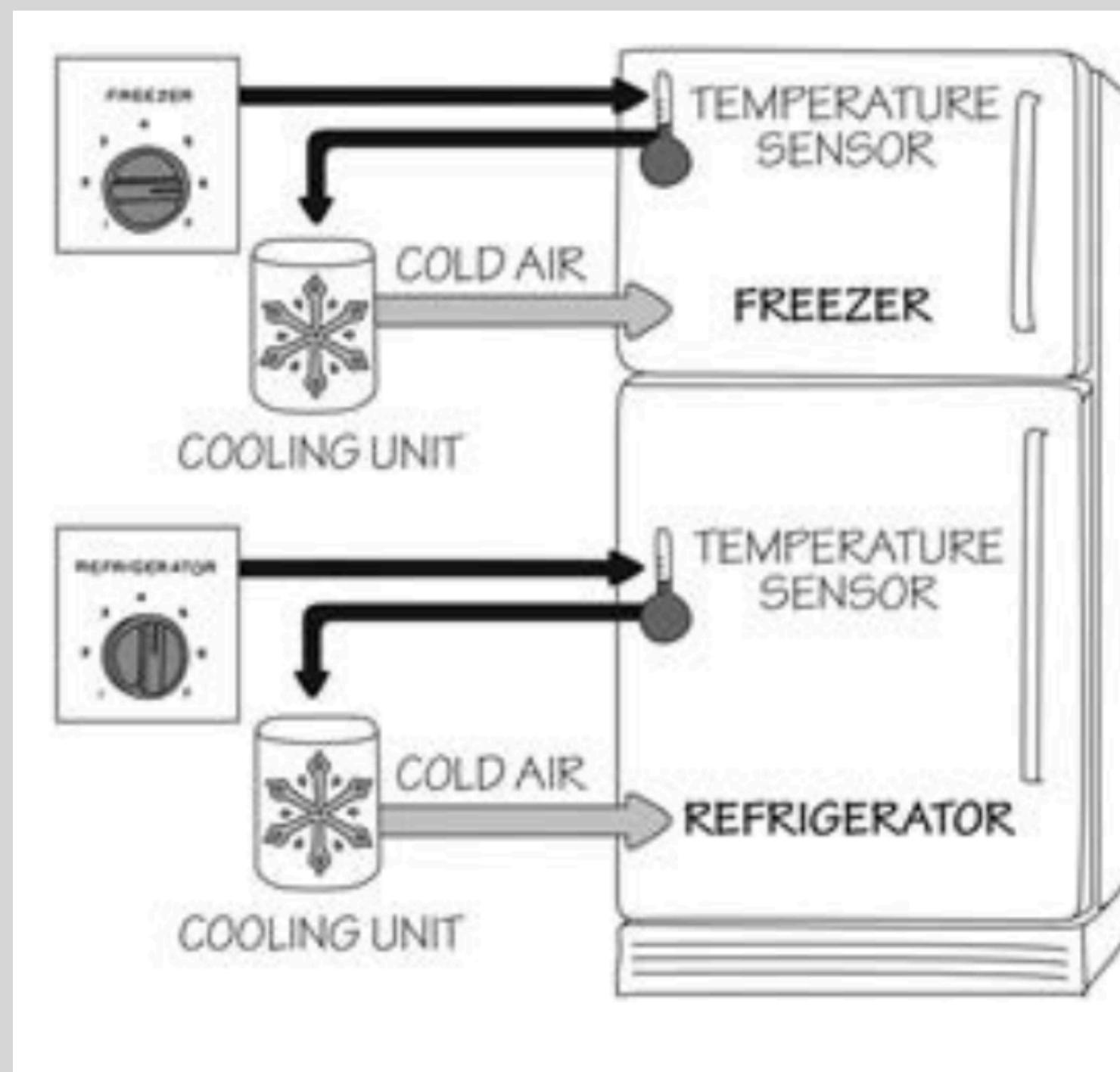
from *The Design of Everyday Things* (1988)

# did the book designer read the book?

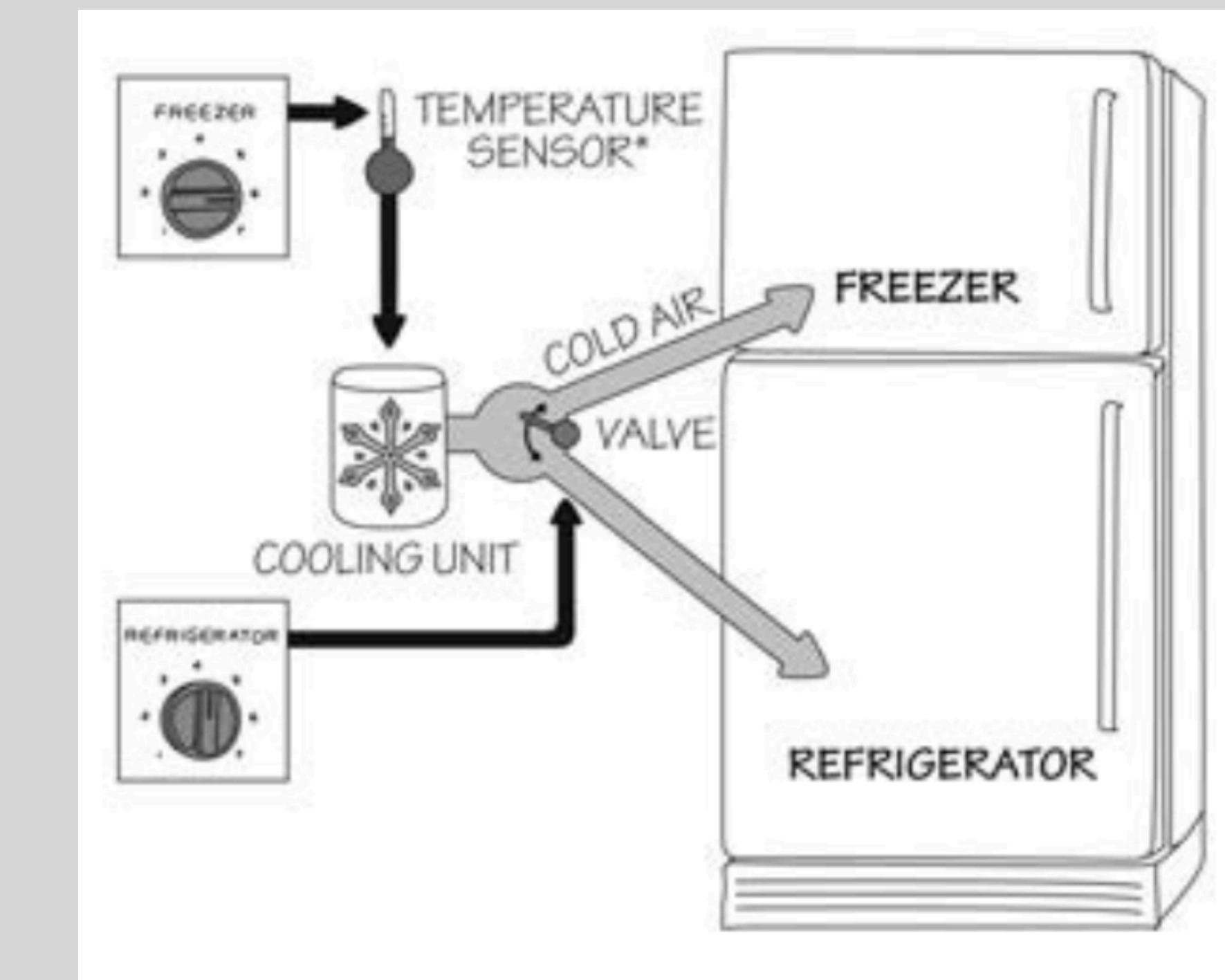




typical controls on American fridge

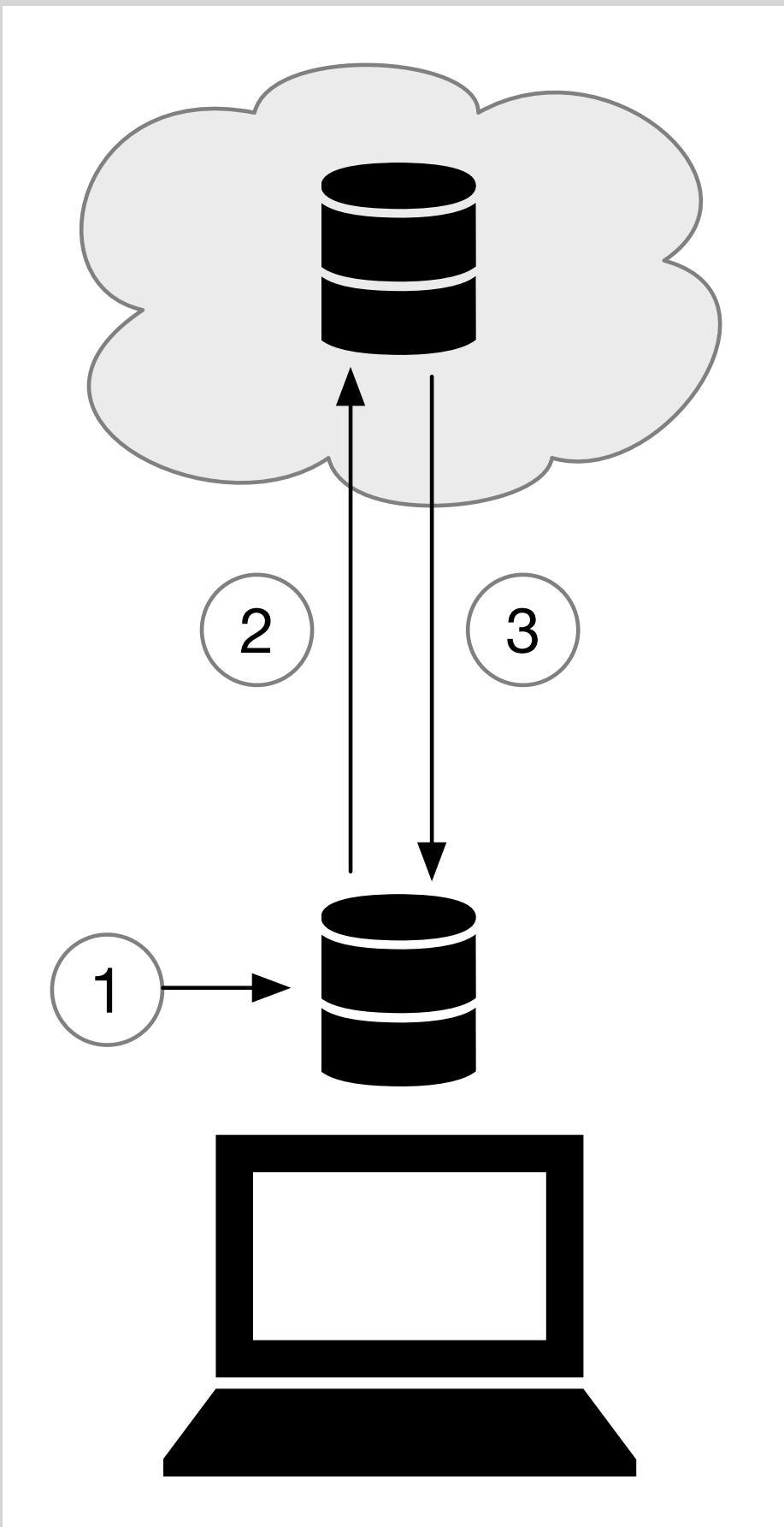


conceptual model (imagined)



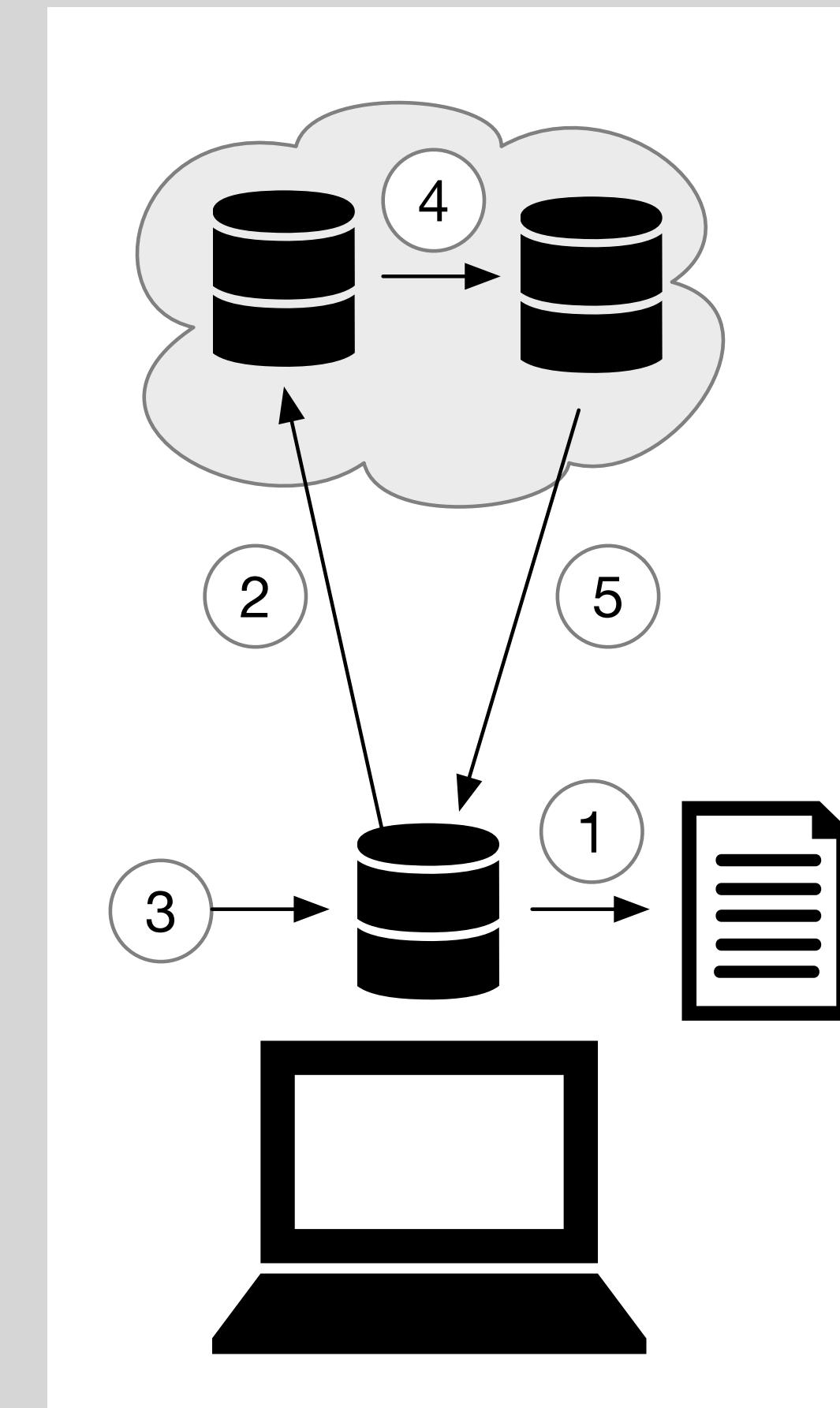
conceptual model (actual)

# Backblaze's conceptual model, imagined and real



**"continuous backup"**  
what I imagined

Each time you modify a file (1), the modification is detected immediately and a snapshot of the new version of the file is taken and copied to the backup server (2), from where it is available for restoring (3).



**"continuous backup"**  
what actually happens

Periodically, the backup utility scans the disk and makes a list of files modified since the last backup (1). It begins to copy files on this list to a special server (2). This process can take a long time, during which you might update additional files (3). When the backup is complete, at some later point the files are copied to a different server (4) from which they can be restored (5).

# a conversation starter

Which captures the role of the conceptual model (CM)? Select all that apply

- (a) The CM is the way an app's user interface presents functionality
- (b) The CM describes user-facing services running in an app's backend
- (c) The CM describes the designer's view of an app's functionality
- (d) The CM describes the user's mental model of an app

another UX puzzle  
Dropbox

**Dropbox:** [Edit](#)

## **Someone accidentally deleted thousands of files in my company Dropbox: how can I quickly undelete them?** [Edit](#)

[Add Question Details](#)[Comment](#) • [Share](#) • [Report](#) • [Options](#)

### **Sharing files with Dropbox (2021)**

A family member of mine wanted to clear space on her drive  
Listed very large files, and deleted ones she didn't recognize  
Panicked message from colleague: where's our data?!



Ava is a party planner

Search AA

## Dropbox

Overview Show ...

<input type="checkbox"/> Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/> Bella Party ☆	2 members	...

does the name change for Ava too?

answer: no, Ava sees no change



Bella is having a party

Search BB

## Dropbox

Overview Show ...

<input type="checkbox"/> Name ↑	Members ▾	⋮ ▾
<input type="checkbox"/> My Party ☆	2 members	...

- Share
- Download
- Send with Transfer
- Request files
- Star
- Rewind
- Rename



Ava is a party planner

Search AA

## Dropbox > Bella Party

Overview Show ...

Name ↑	Members ↓	...
<input type="checkbox"/> Party Plan ☆	2 members	...

what about this case? folder inside shared folder

answer: yes, name changes for Ava too



Bella is having a party

Search BB

## Dropbox > Bella Party

Overview Show ...

Name ↑	Members ↓	...
<input type="checkbox"/> Party Plan ☆	2 members	...

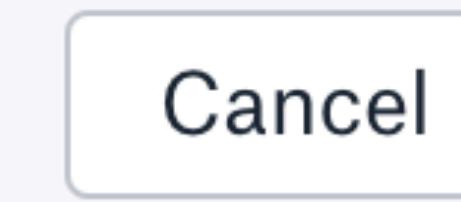
- Share
- Download
- Send with Transfer
- Request files
- Star
- Rewind
- Rename



## Remove shared folder?

Bella deletes  
shared folder  
Bella Party

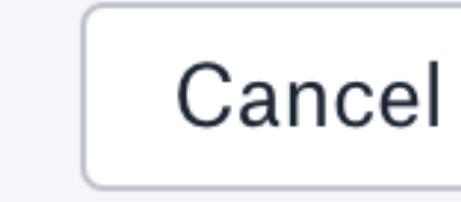
Are you sure you want to remove the shared folder **Bella Party** from your Dropbox? This folder will stay shared with any existing members. You can re-add it later.

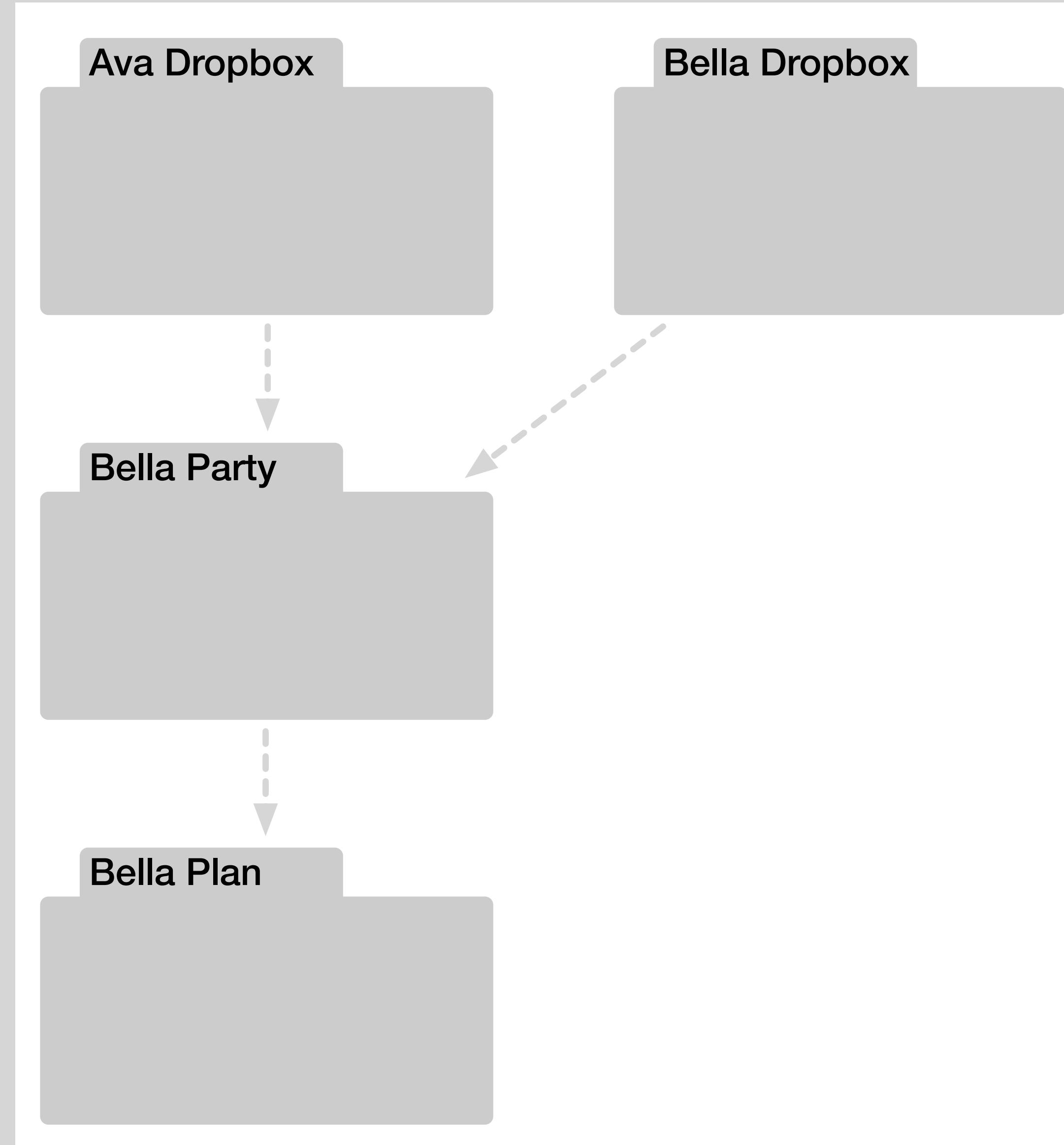
  
Cancel  
Remove

## Delete folder?

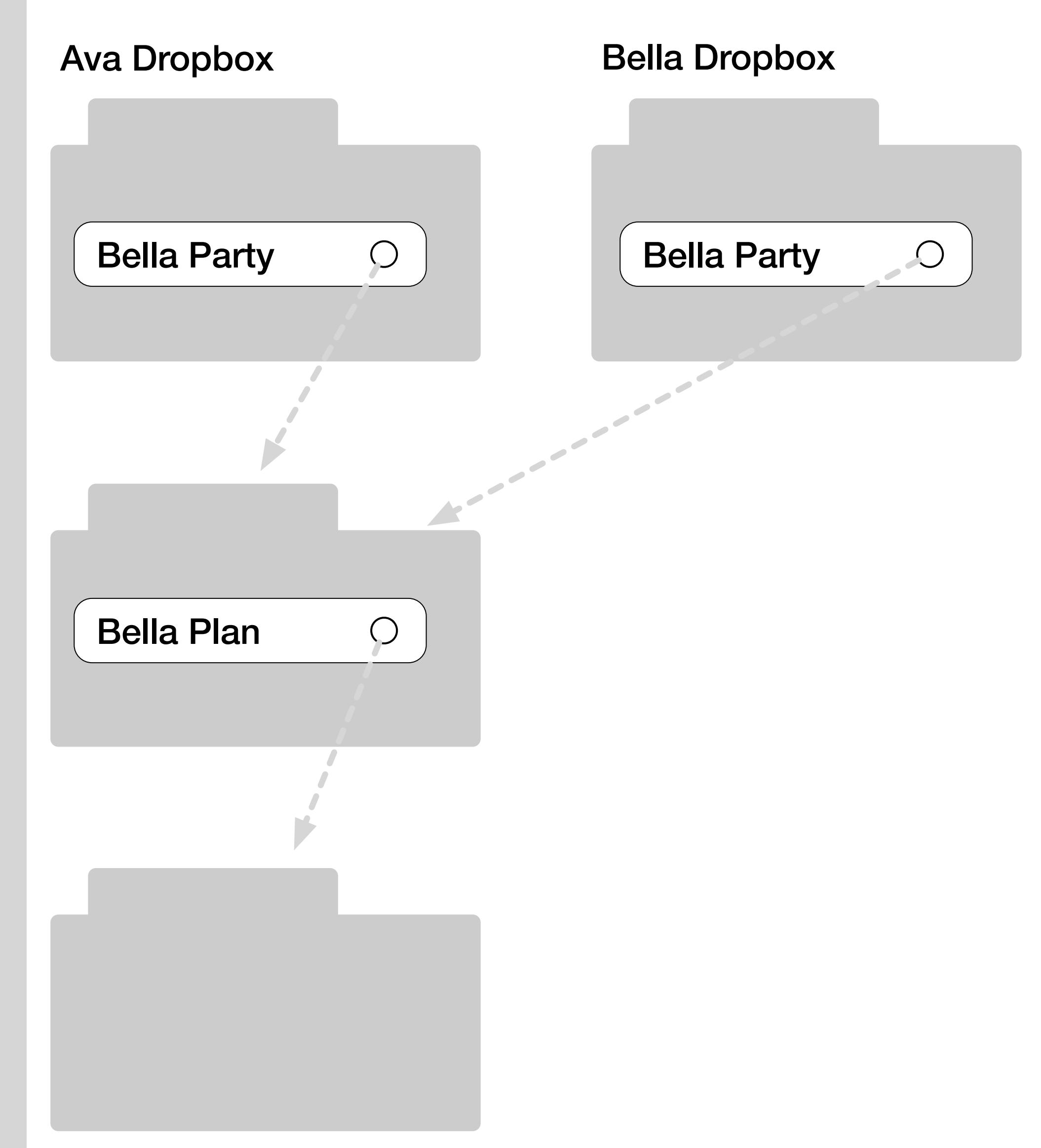
Bella deletes  
Bella Plan from  
shared folder  
Bella Party

Are you sure you want to delete **Bella Plan** from the shared folder 'Bella Party'?

  
Cancel  
Delete

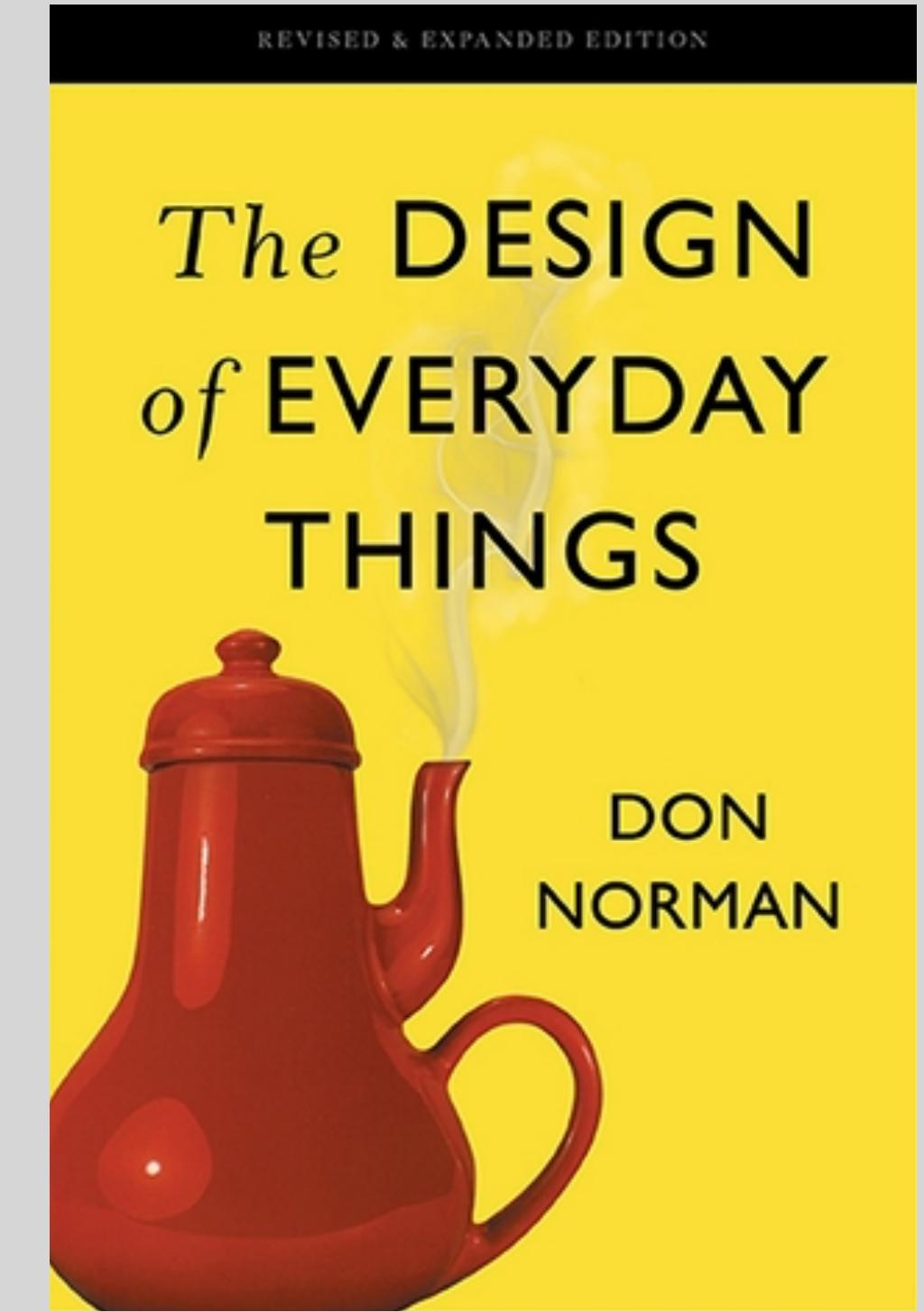
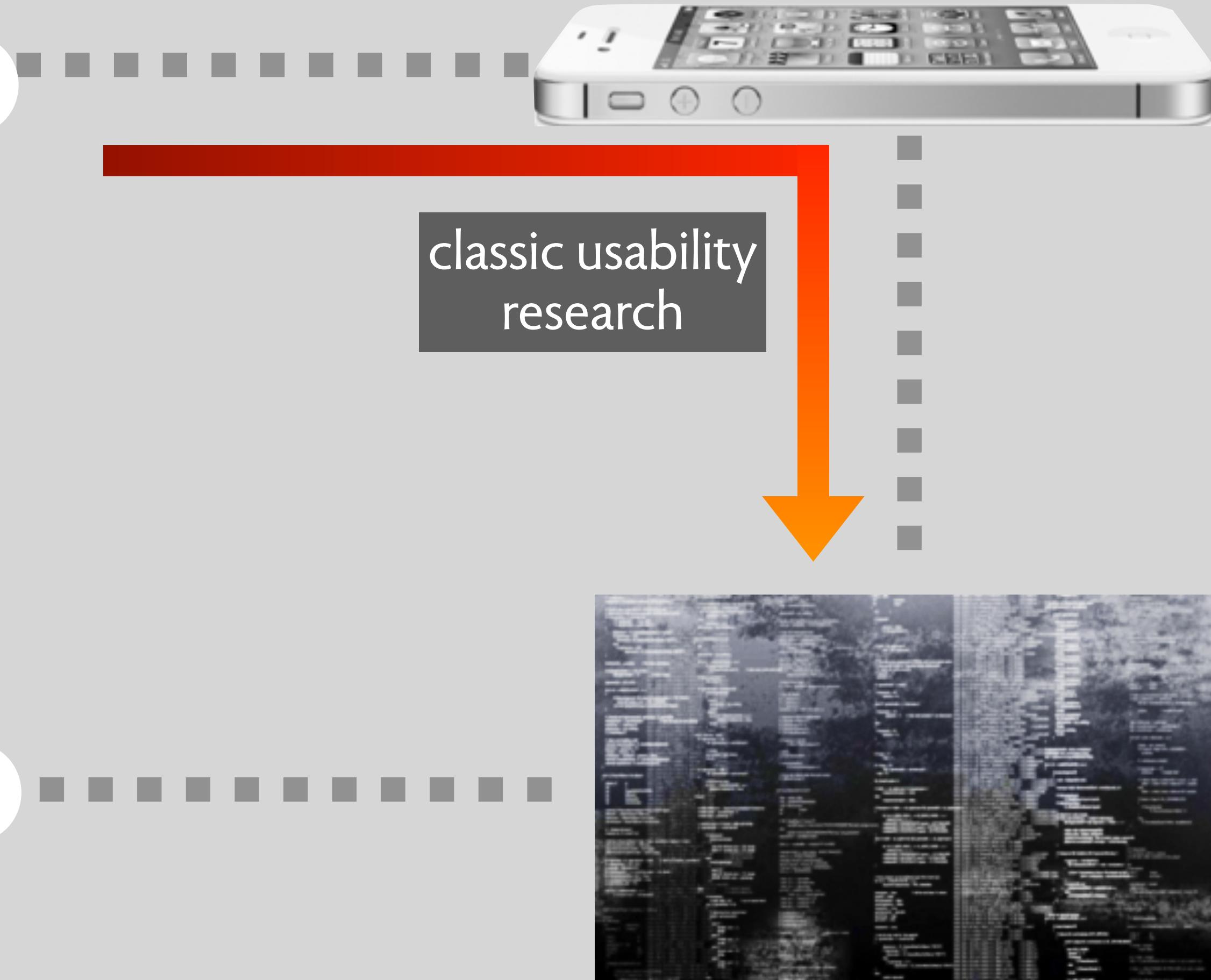
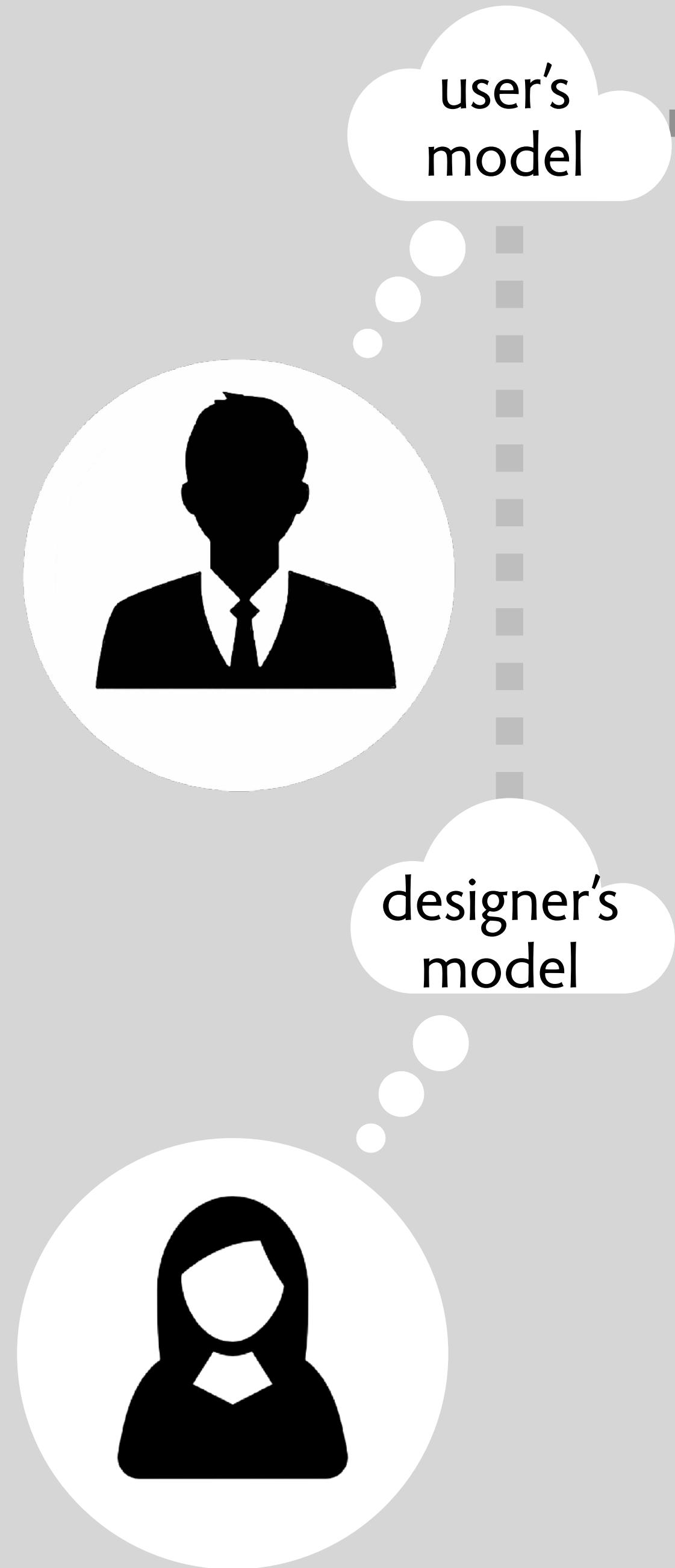


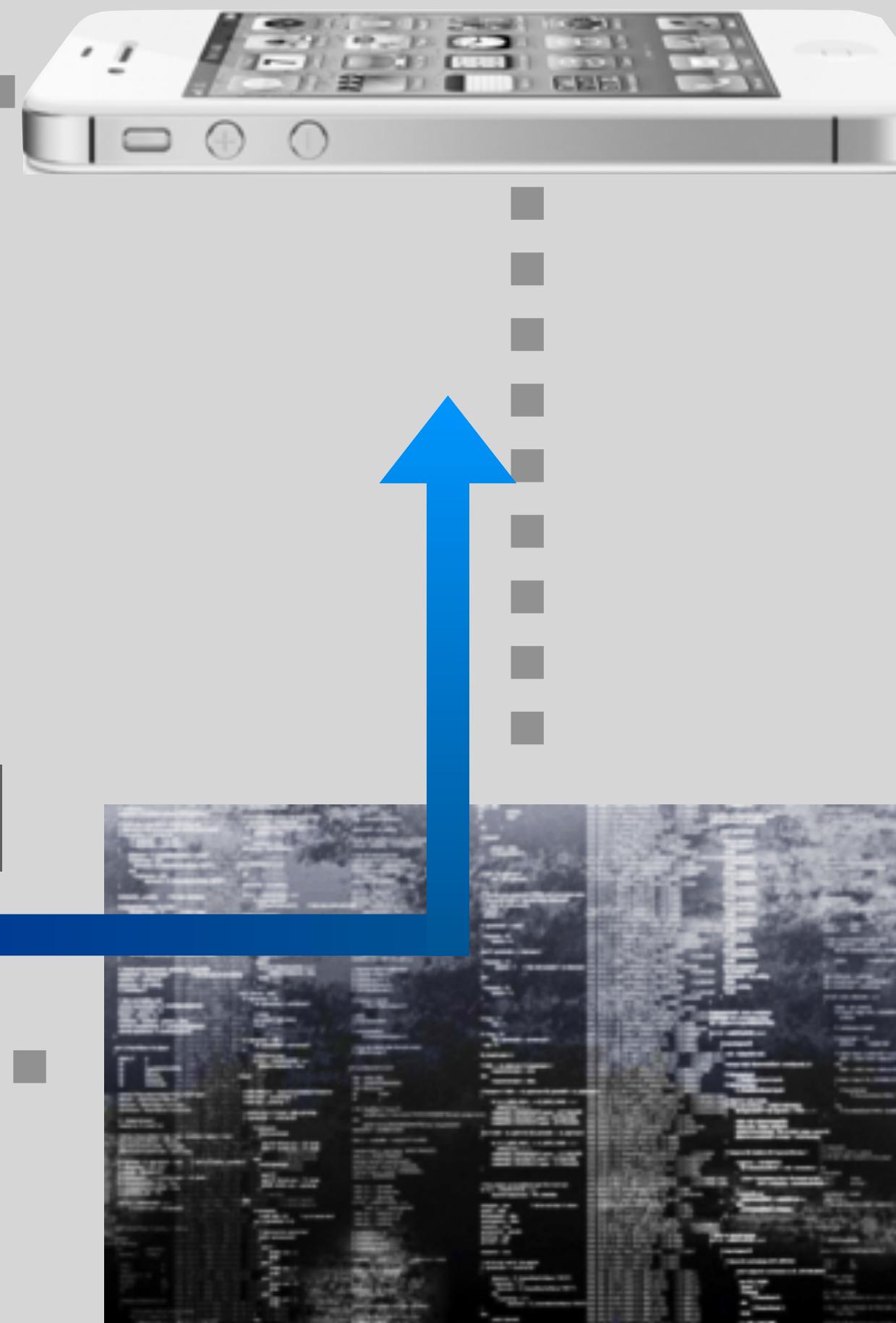
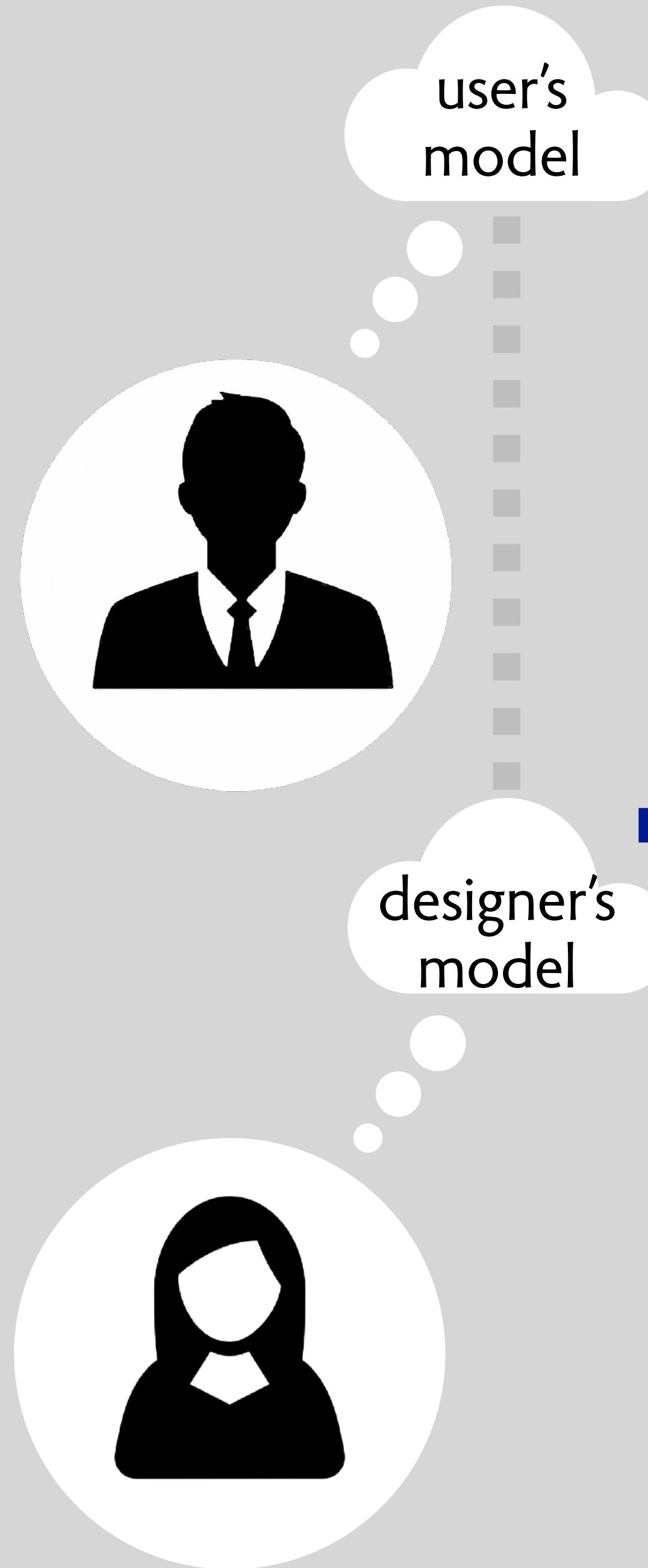
how many users believe the folder concept works



how folders actually work (in Dropbox, Unix, Multics)

conceptual models:  
explanation or design?





but what if the designed  
model is just wrong?

## user-centered design (1980s)

concepts are a **byproduct** of design

designer's job: **shape UI** to project concepts

concepts are **psychological**

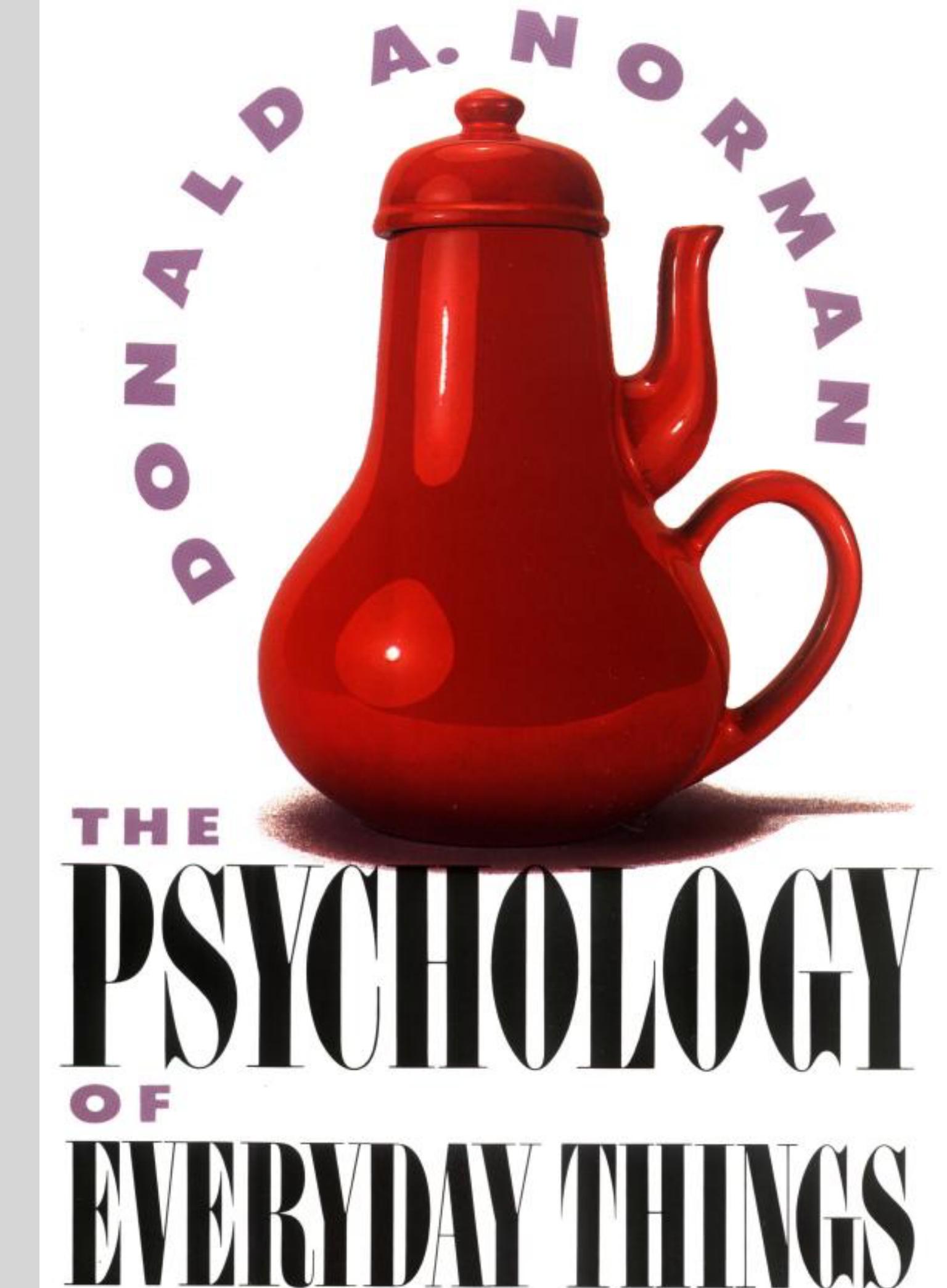
## concept-based design

concepts are the **essence** of design

designer's job: **shape concepts**

concepts are **computational**

Copyrighted Material



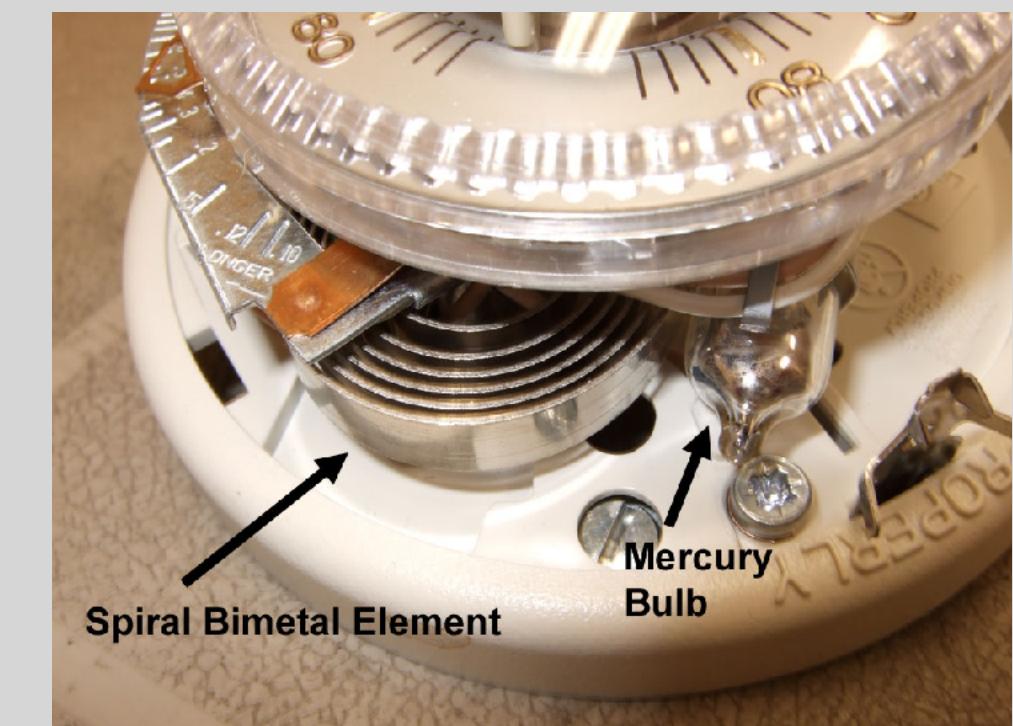
Copyrighted Material

# implications for UX designers



## classic UX approach

underlying mechanism is fixed by engineers  
UX designer's job is to create an **explanation**  
UX is secondary to engineering



[electrical-forensics.com](http://electrical-forensics.com)



## new UX approach

the conceptual model is the mechanism!  
UX designer's job is to create **the model**  
UX and engineering go hand-in-hand

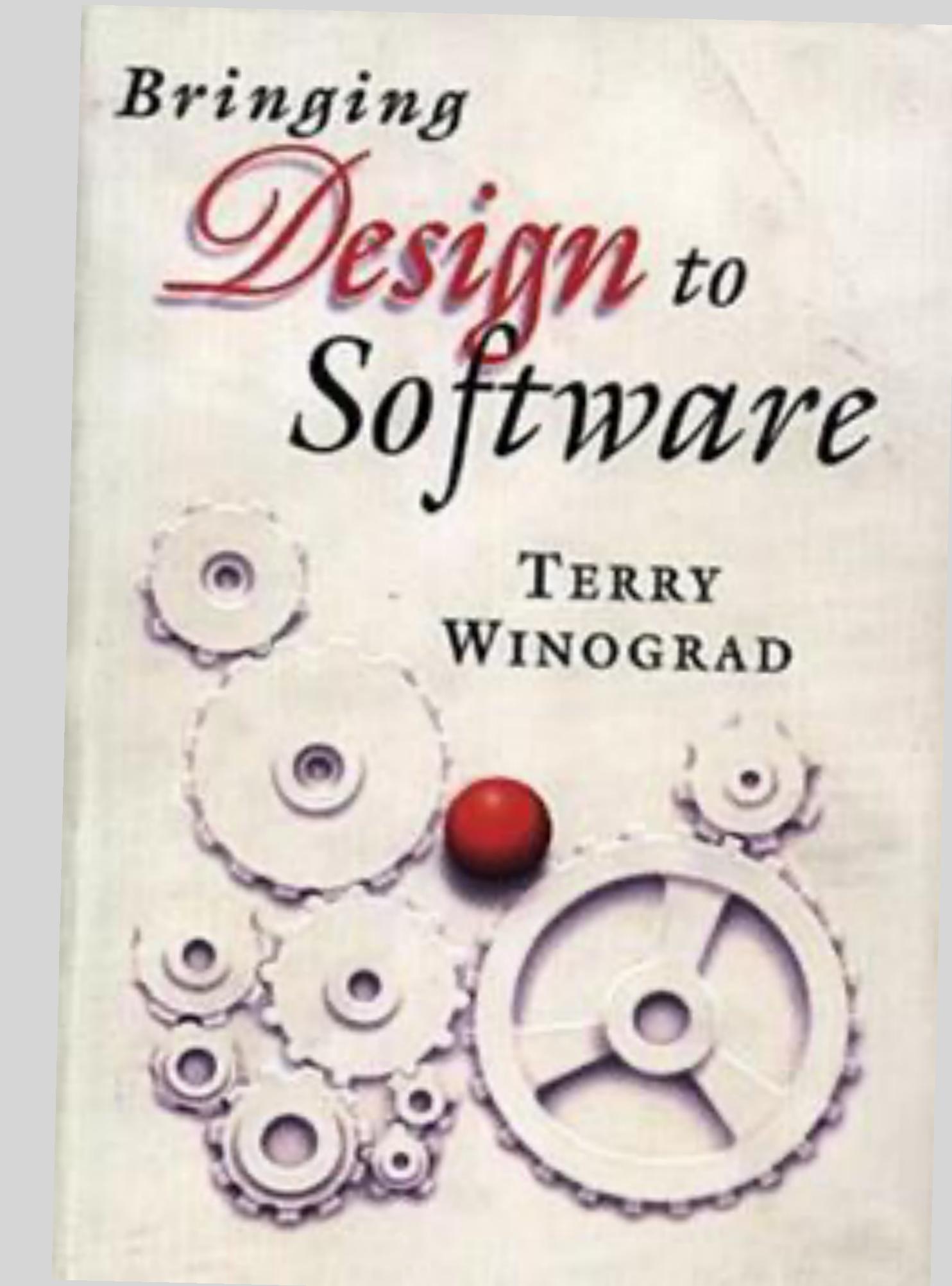


When you go to design a house  
**you talk to an architect first**, not  
an engineer. Why is this?

Because the criteria for what  
makes a good building fall outside  
the domain of engineering.

Similarly, in computer programs,  
**the selection of the various  
components** and elements of the  
application must be driven by the  
conditions of use. **How is this to  
be done?** By software designers.

*A Software Design Manifesto*  
Mitchell Kapor, 1996  
paraphrasing slightly



# a conversation starter

Which are implications of the claim that concepts are designed? Select all that apply

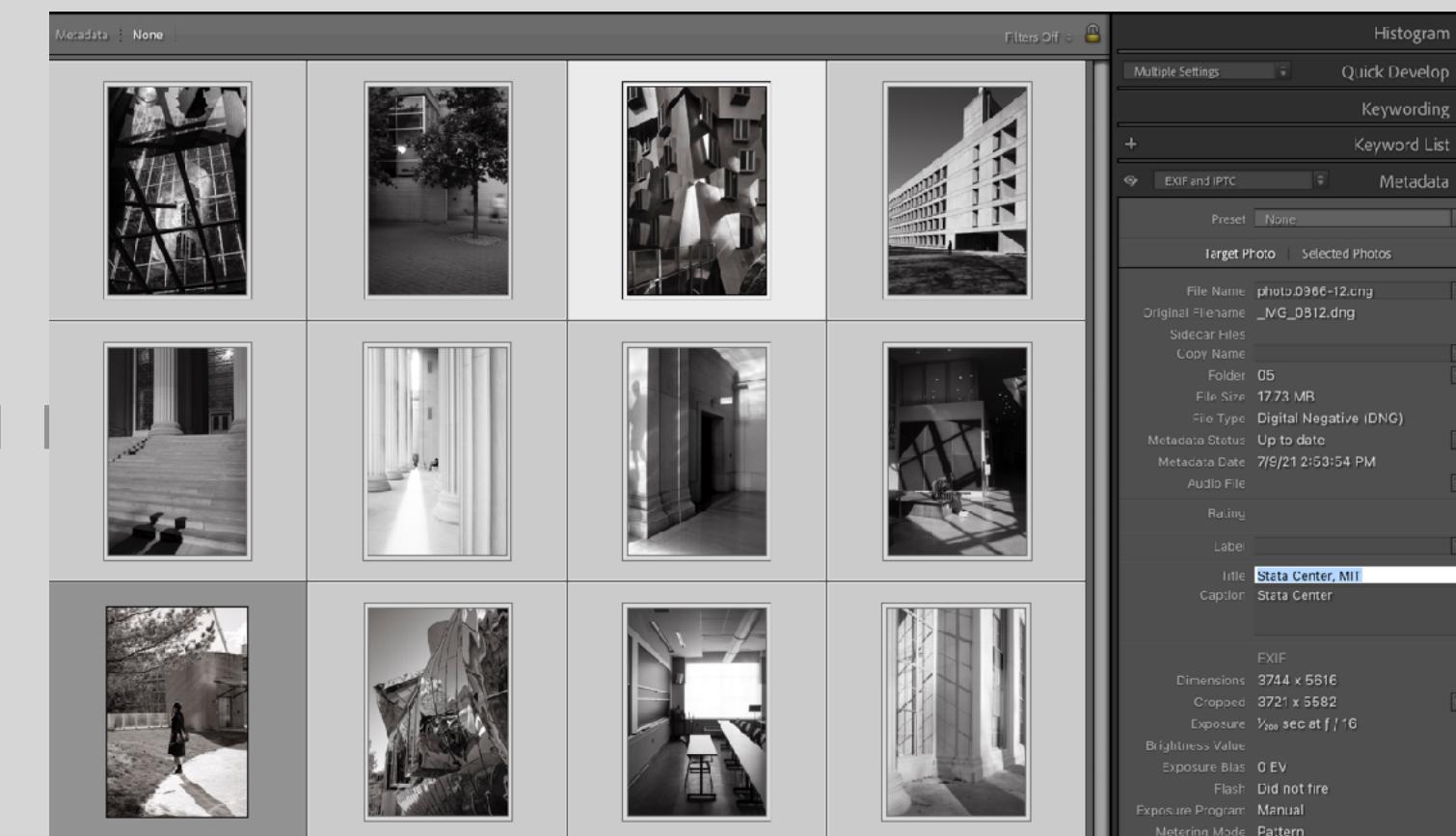
- (a) UX architects and engineers must collaborate on the design of services
- (b) A well-designed UI can mitigate a bad concept design
- (c) Novelty now matters more than familiarity in concept design

why concepts?  
finding granularity

## a limitation of UI-driven UX

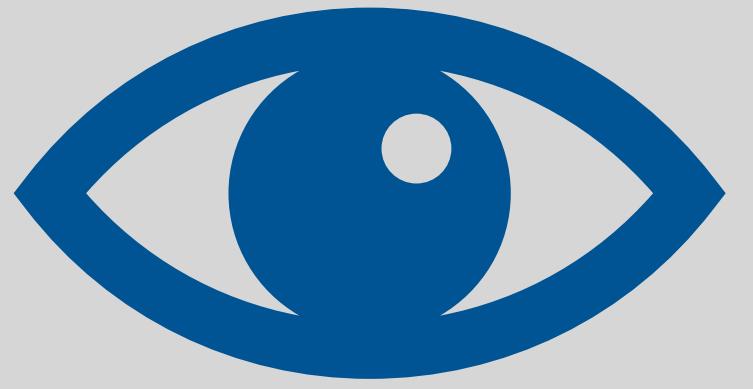
if the user interface is the focus of our attention,  
how can we ask if it projects the right concepts?  
and what if the underlying concepts are wrong?

designer's  
concept

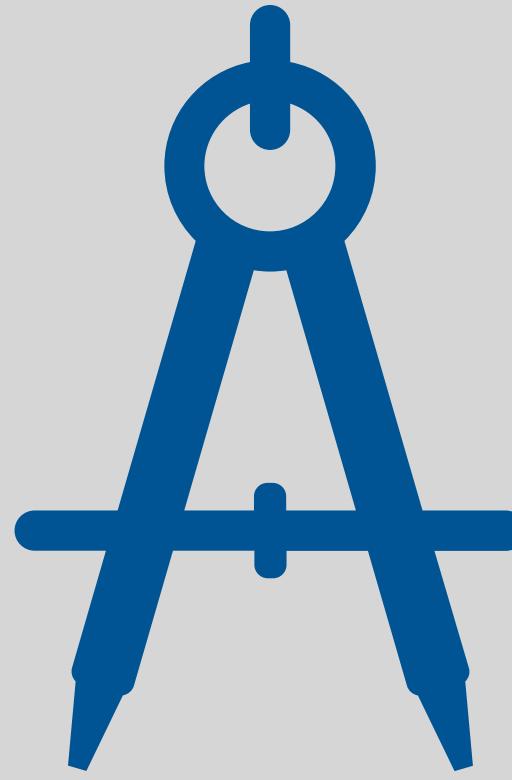


user's  
concept

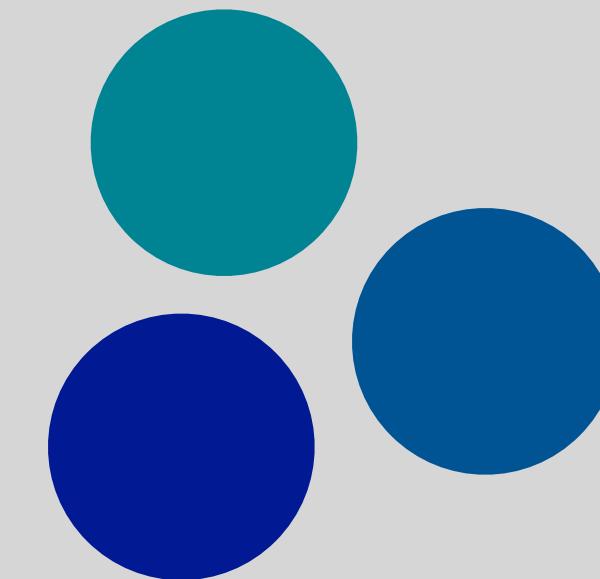
# essential features of a conceptual model



**user facing**  
what the user experiences  
no invisible implementation



**clear & precise**  
know what it means & implies  
can specify details if we want

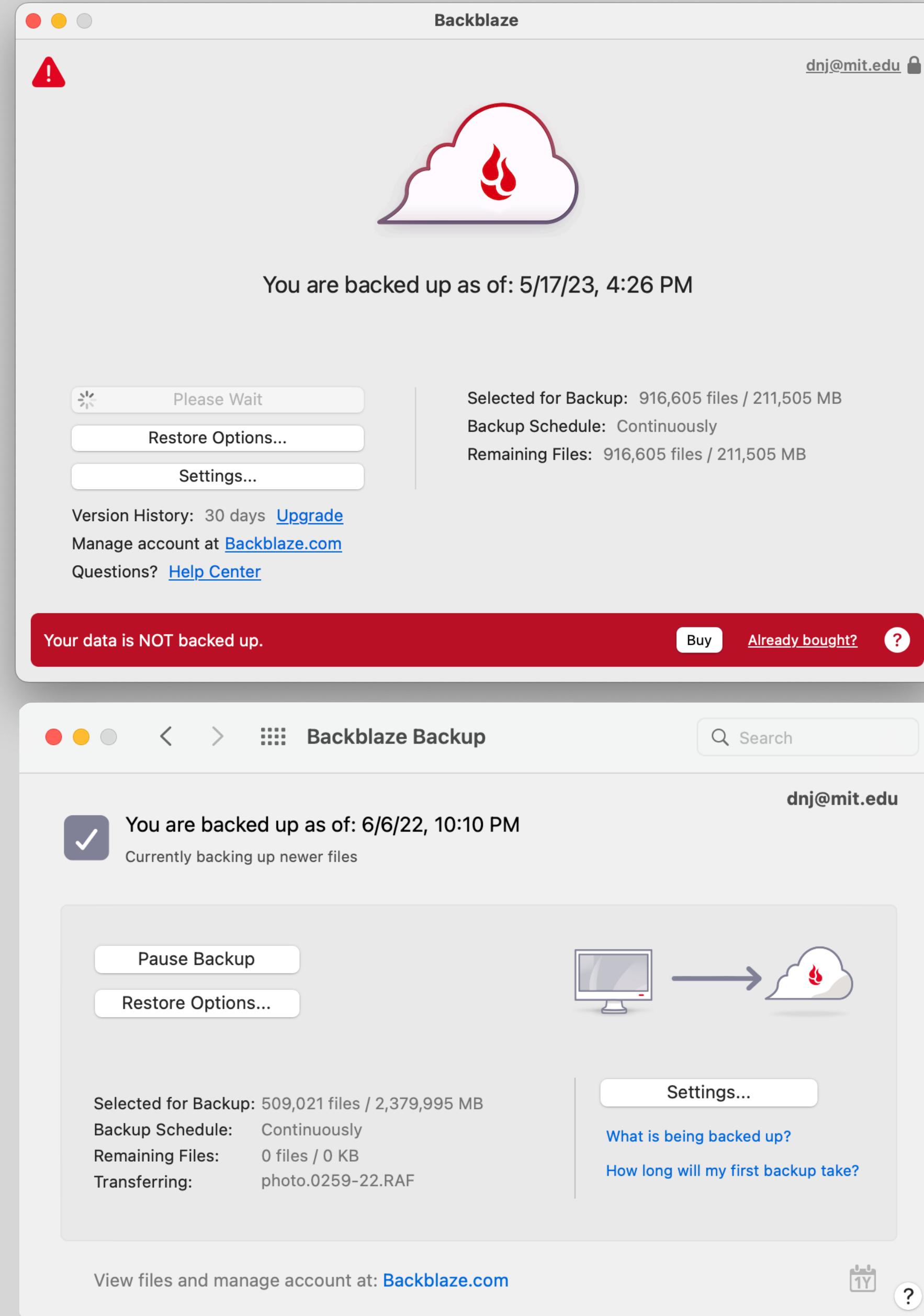


**granular**  
separable components  
independently grasped

**why granularity matters**  
incremental work  
division of labor  
reuse of prior knowledge  
familiarity for users

# explaining Backblaze with concepts

Periodically, the backup utility scans the disk and makes a list of file modified since the last backup (1). It begins to copy files on this list to a special server (2). This process can take a long time, during which you might update additional files (3). When the backup is complete, at some later point the files are copied to a different server (4) from which they can be restored (5).



*a scenario that conflates concepts*

Backblaze is a **paid service** (so when service period has expired, may still see results of previous periods)

Backblaze uses a **write-only filestore** to maintain backups of your files

To backup the files, Backblaze runs a periodic batch task that creates a **worklist** of modified files.

*separating out concepts*

so what's a concept?  
defining concepts

## ▲ Jackson structured programming (wikipedia.org)

post

106 points by haakonhr 63 days ago | hide | past | favorite | 69 comments

session

upvote

favorite

▲ danielnicholas 63 days ago [-]

user: danielnicholas

created: 63 days ago

karma: 11

You might find helpful an annotated version [0] of Hoare's explanation of JSP that I edited for a Michael Jackson festschrift

; I'd point to these ideas as worth knowing:

ing problem that involves traversing structures can be solved very systematically. HTDP addresses this class, but bases one structure only on input structure; JSP synthesized it.

comment

- The karma one archetypal problems that, however you code, can't be pushed under the rug—most notably structure clashes—and just recognizing them

- Coroutines (or code transformation) let you structure code more cleanly when you need to read or write more than one structure. It's why real iterators (with yield), which offer a limited form of this, are (in my view) better than Java-style iterators with a next method.

- The idea of viewing a system as a collection of asynchronous processes (Ch. 11 in the JSP book, which later became JSD) with a long-running process for each real-world entity. This was a notable contrast to OOP, and led to a strategy (seeing a resurgence with event storming for DDD) that began with events rather than objects.

[0] <https://groups.csail.mit.edu/sdg/pubs/2009/hoare-jsp-3-29-09...>

▲ ob-nix 63 days ago [-]

... this brings back memories! In the late eighties I, as a teenager, found a Jackson Struct. Pr. book at the town library. I remember I was amazed at the text and wondered why I hadn't heard about the method before.

If I remember correctly did the book clearly point out backtracking as a standard method, while mentioning that most languages lacked that, so it had to be implemented manually.

▲ CraigJPerry 63 days ago [-]

This is referenced(1) as a core inspiration in the preface to "How to Design Programs" but i never researched it further because i've found the "design recipes" approach in htdp to be pretty solid in real life problems

# a sample concept: Upvote

 **Hacker News** new | past | comments | ask | show

 ▲ GPT-5 (openai.com)

2929 points by rd 22 hours ago | hide | past | favorite | 2298 comments

[https://www.youtube.com/watch?v=0Uu\\_VJeVVfo](https://www.youtube.com/watch?v=0Uu_VJeVVfo)



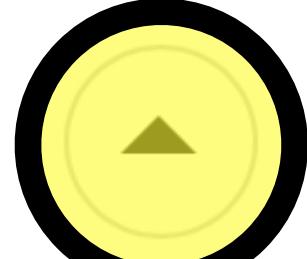
**One more for dinner**

Charlottesville | 3h ago

Again, please. He didn't take this position for our benefit. He did it for his.

[Reply](#) 65 [Recommend](#) [Share](#)

Flag



4

Can any one here explains how the transitive closure operator works in Alloy in terms of the matrix. I mean what's translation rule for translating closure operator into actual matrix operation.



alloy



*what's a concept?*

a coherent **unit** of behavior

**user-facing** (a behavioral pattern)

a nano **service** (a backend API)

**reusable & familiar**

designed, coded and explained **independently**

# defining a concept

**concept** Upvote [User, Item]

**purpose** rank items by popularity

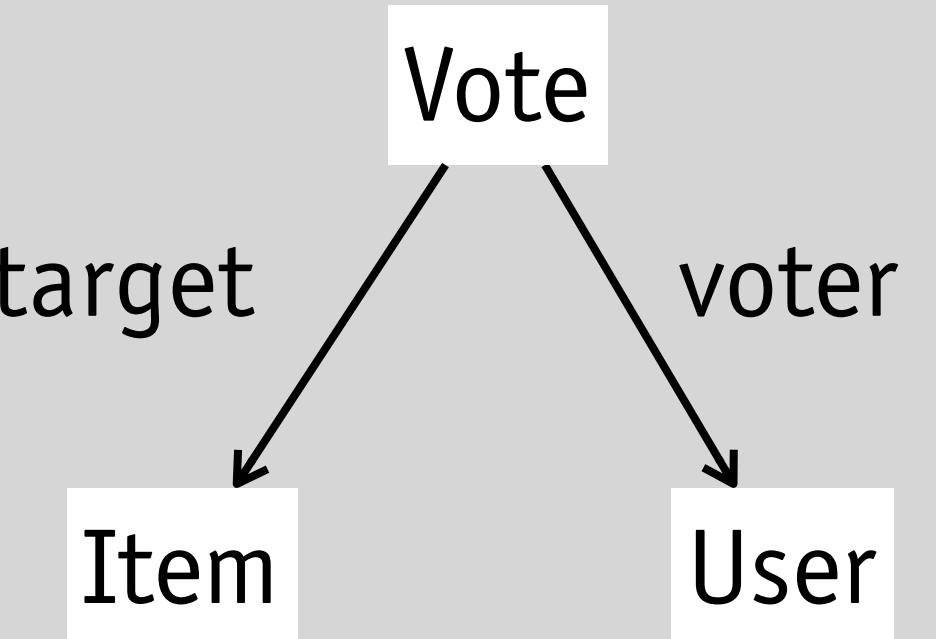
**principle** after series of votes of items, the items can be ranked by their number of votes

**state**

a set of Votes with  
a voter User  
a target Item

**actions**

upvote (user: User, item: Item)  
unvote (user: User, item: Item)



# similar UIs, different concepts

## concept Upvote

**purpose** rank items by popularity

**principle** after series of votes of items, the items can be ranked by their number of votes

This is homework and I'm having a  
are the definitions of the objects:

8

▼

★

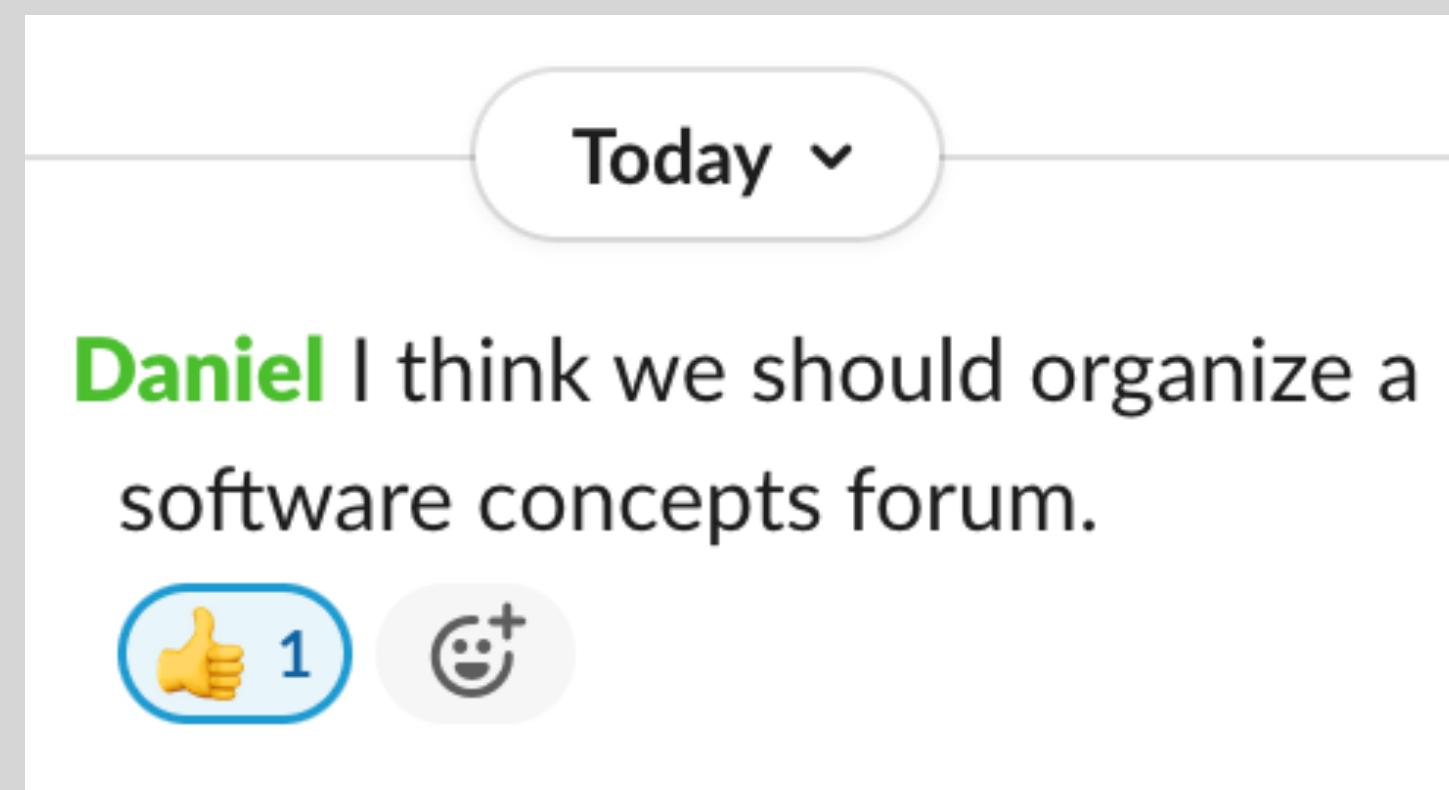
1

```
sig Library {  
    patrons : set Person,  
    on_shelves : set Book,  
}
```

## concept Reaction

**purpose** send reactions to author

**principle** when user selects reaction, it's shown to the author (often in aggregated form)



## concept Recommendation

**purpose** use prior likes to recommend

**principle** user's likes lead to ranking of kinds of items, determining which items are recommended



# extreme decoupling (xd): syncs & polymorphism

*how to ensure concepts are independent?*

**no calls** from one to another

**no assumptions** about external **types**

*suppose we want to notify authors when their posts are upvoted*

*we have Upvote.upvote (...) and Notification.notify (...)*

```
sync NotifyAuthorOnUpvote  
when Upvote.upvote (item)  
where author of item is user in Post concept  
then Notification.notify (user, item + " upvoted")
```

**concept** Upvote [User, Item]

**state**

a set of Votes with  
  a voter User  
  a target Item

# concepts as carriers of design knowledge

**concept:** Upvote

**related concepts**

Rating, Recommendation, Reaction, ...

## design variants

downvote as unvote  
use age in ranking  
weigh downvotes more  
various identity tactics  
freezing old posts

## typical uses

social media posts  
comments on articles  
Q&A responses



## known issues

high votes can promote old content  
feedback favors early upvotes  
upvoting encourages echo chamber  
preventing double votes

**often used with**

Karma, Auth, ...

# a first concept exercise

**concept** Bookmarking

**purpose** easy access to favorite links

**principle** after saving several urls, a user can view them all (and then easily pick one and revisit it)

**state**

a set of Bookmarks with  
a usernameString  
a url String

**actions**

```
save (username: String, url: String)  
// create a new bookmark  
// with this username and url
```

**adding labels**

extend the concept definition so that bookmarks have user-defined labels

**adding actions**

suggest a couple of additional actions

**exploring genericity**

does this concept exploit any details of usernames or urls? should it?

takeaways

**conceptual models**  
shaping UIs to teach designer's model

**concept design**  
shaping the designer's model

**concepts**  
granular, purposeful, behavioral

**concept parts**  
name, purpose, principle, state & actions