

Conceptual Modeling

Interface Pessoa-Máquina - 25/26 - LEI / UM

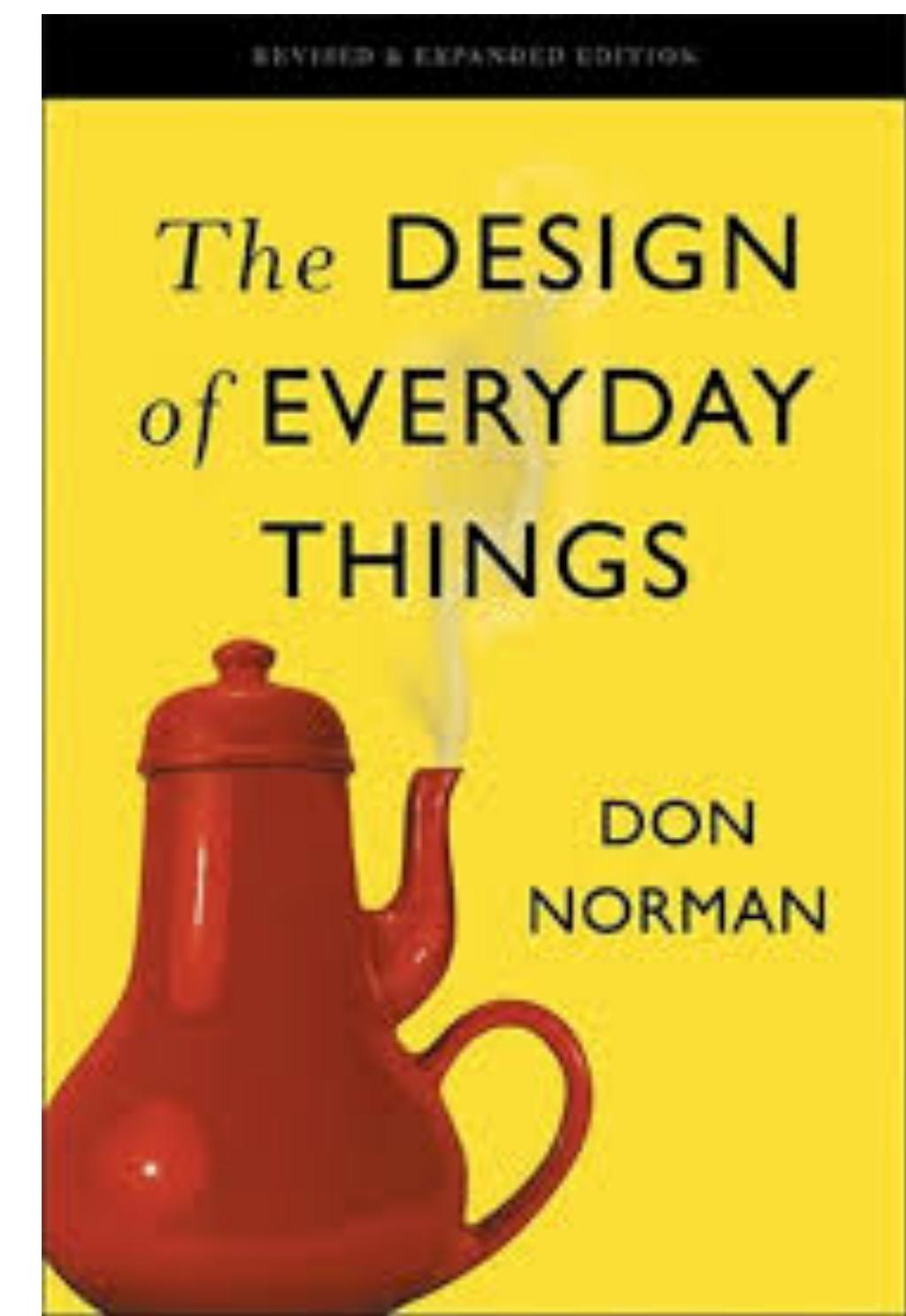
Hugo Pacheco

hpacheco@di.uminho.pt

Modelo mental



- Um **modelo mental** é a representação interna do utilizador de como um sistema funciona
- Já vimos que construir um modelo mental é essencial
 - Learnability ⊂ Usability
 - Match between System and Real World
- Também denominado “**user's conceptual model**”
- “[...] you [**the user**] form a **conceptual model** of the device and mentally simulate its operation. You can do the simulation because the parts are visible and the implications clear.”

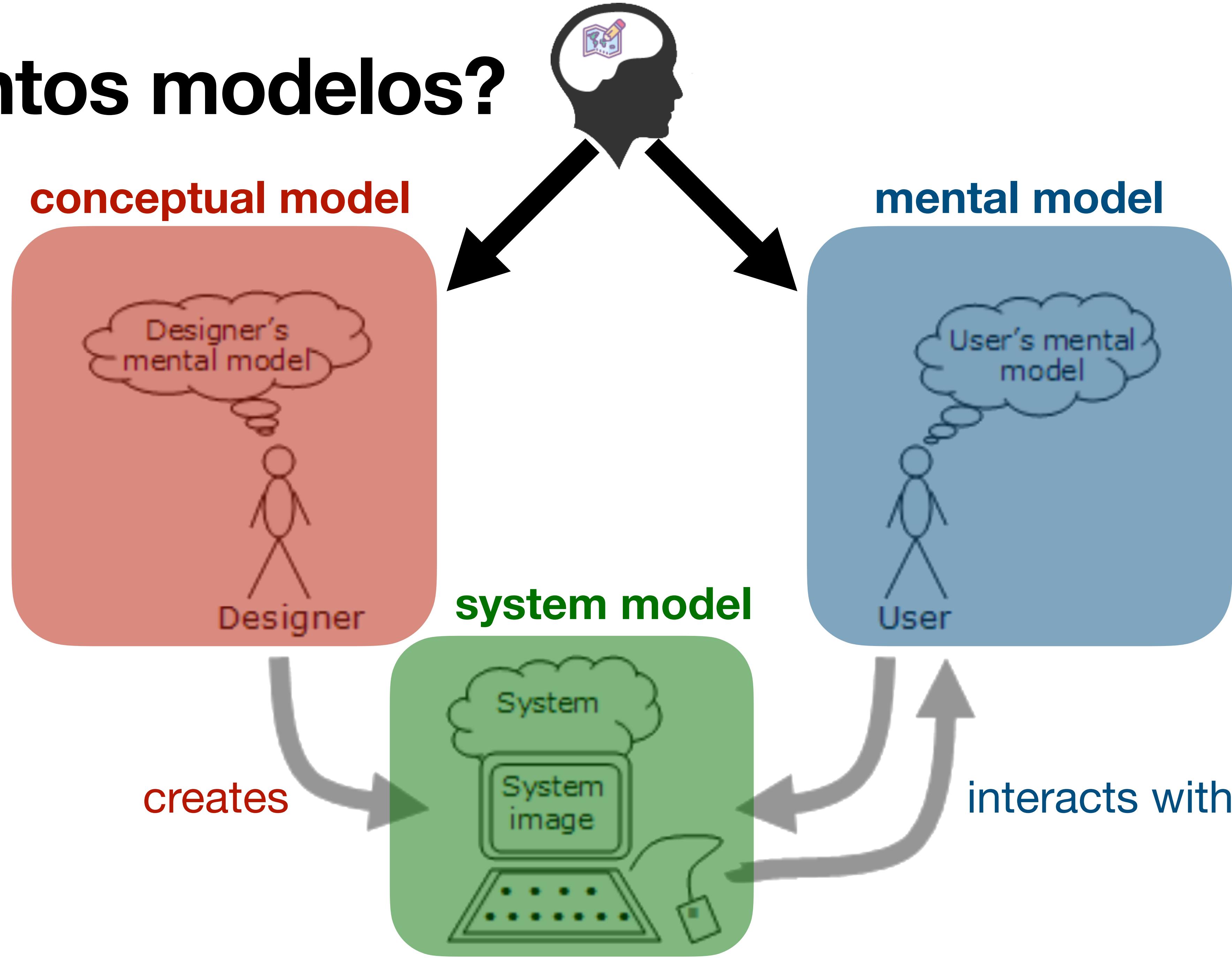


Modelo mental

- Essencial para
 - Determinar causas e efeitos
 - Suportar explicações
 - Fazer previsões
 - Compreender dispositivos similares



Quantos modelos?



Quantos modelos?

- **System model**: modelo de como o sistema realmente funciona
 - **System image**: o que o utilizador vê do **system model**
- **Mental model**: interpretação do utilizador de como o sistema funciona
 - Desenvolvido por interação com o sistema
 - Objetivo: replicar o **system model**
- **Conceptual model**: modelação do designer/developer de como o sistema funciona
 - Desenvolvido construindo a **system image**, que abstrai o sistema real
 - Objetivo: assistir o utilizador a criar um bom **mental model**

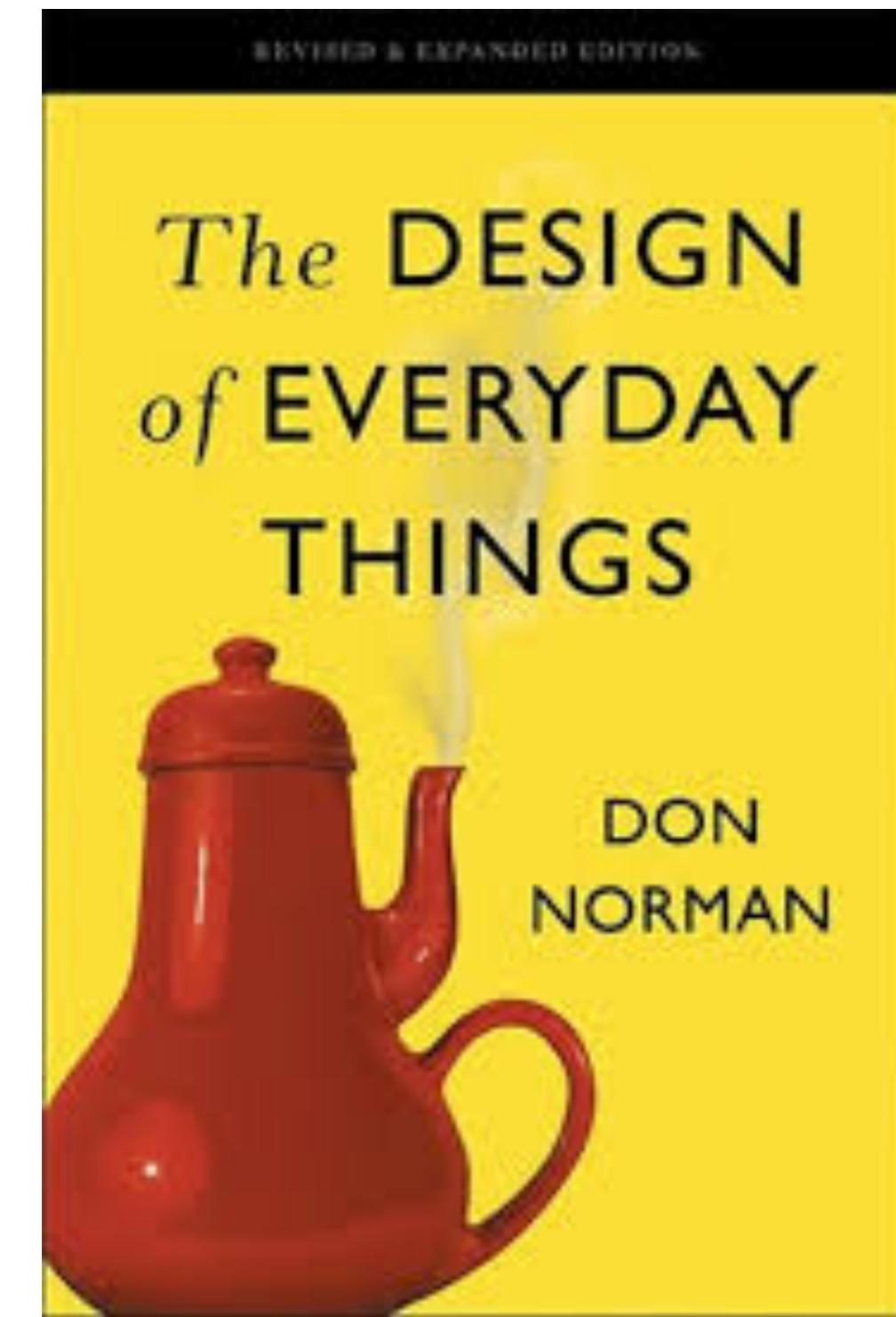
Modelo mental !

- Incompleto ou inexato (pode conter erros ou medidas incertas)
- Instável ou simplista (utilizador esquece-se facilmente de detalhes e pode ignorar fenómenos complexos)
- Não científico (utilizador pode ter comportamentos “supersticiosos”)
- Em constante evolução
- Sem limites claros (utilizador pode confundir dispositivos ou operações similares)
- Impulsivo (utilizador pode tentar executar ações erradas em vez de planear)



Modelo conceptual

- Fundamental principles of designing for people
 1. provide a good **conceptual model**
 - **Affordances**: dicas de como operar o objeto
 - **Mappings**: relações entre propriedades do objeto e conceitos do modelo
 - **Constraints**: restrições ao uso do objeto
 - **Feedback**: dar informação do que está a ser ou foi feito
 2. make things visible
 - Reforçar os outros princípios, e.g., perceived affordances



Modelo conceptual

- **Affordances**



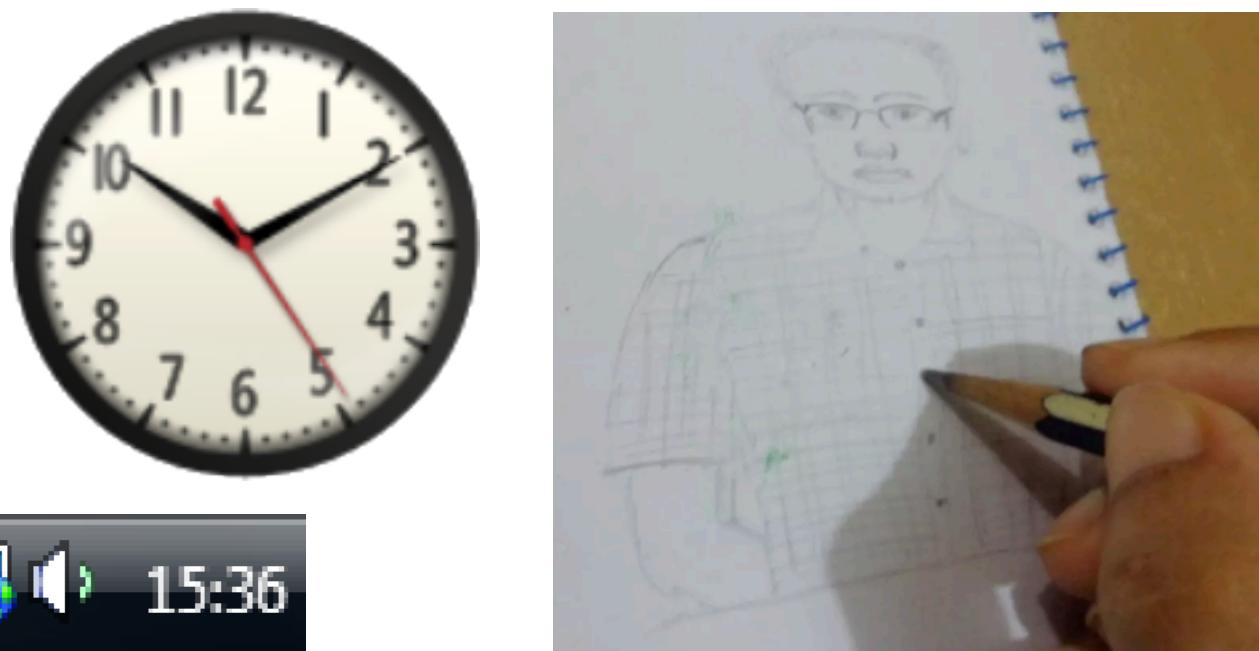
- **Mappings**



- **Constraints**

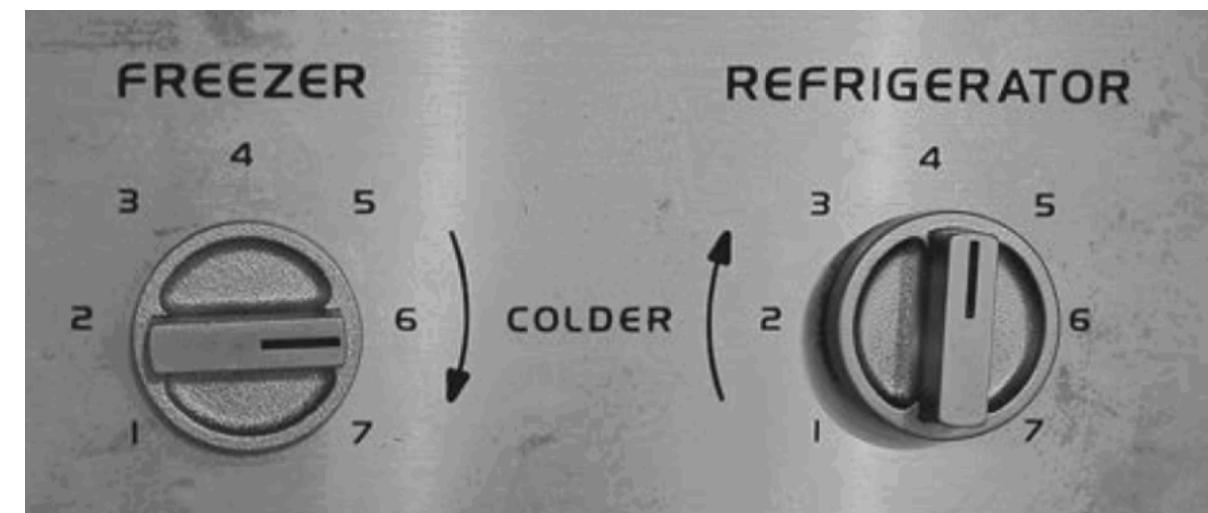


- **Feedback**

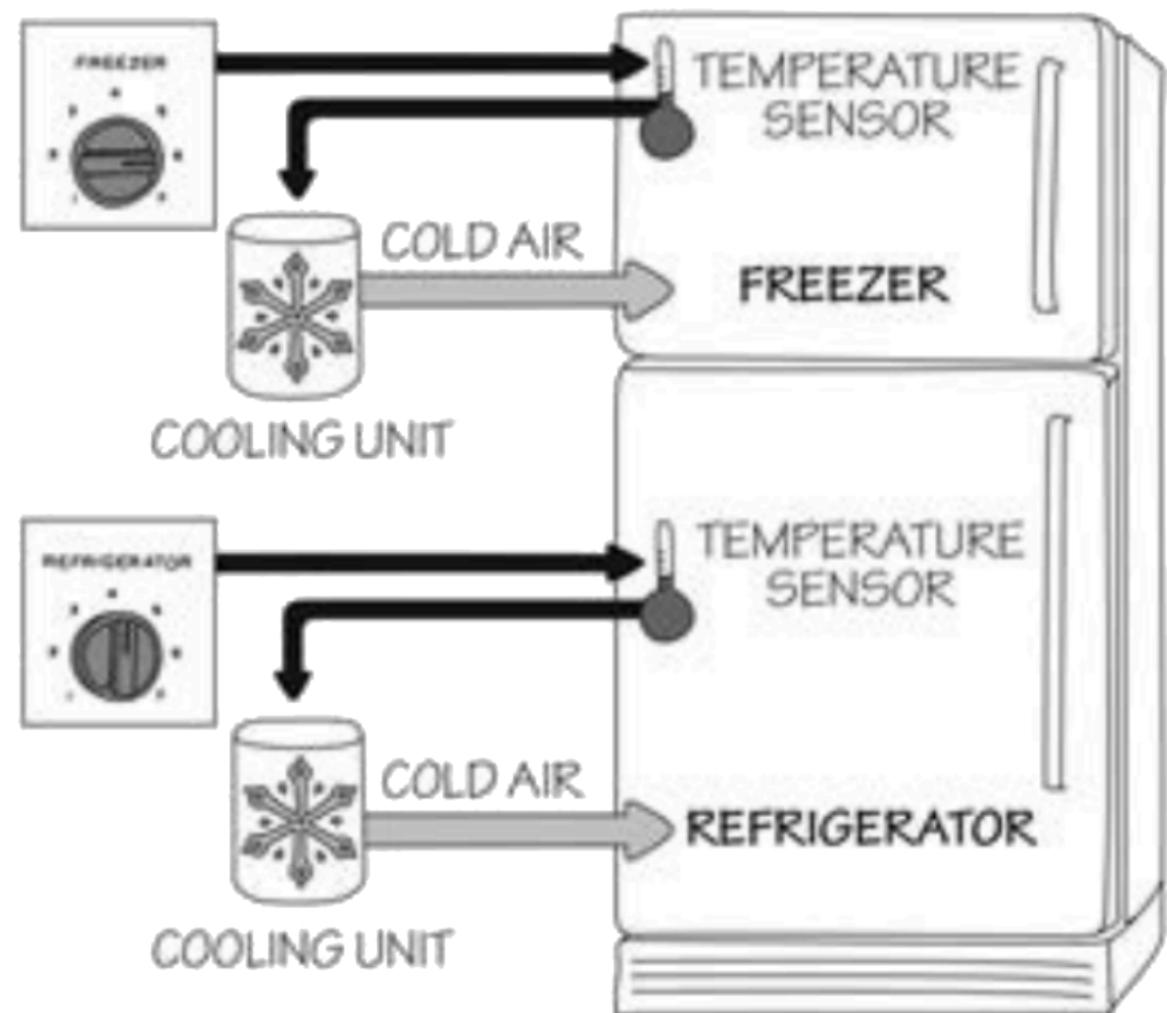


- **Affordances:** buracos para inserir algo
- **Mappings:** sugestão de buracos para inserir dedos
- **Constraints:** buraco grande estreito para vários dedos, buraco pequeno redondo para polegar
- **Feedback:** movimento das lâminas ao mover os dedos

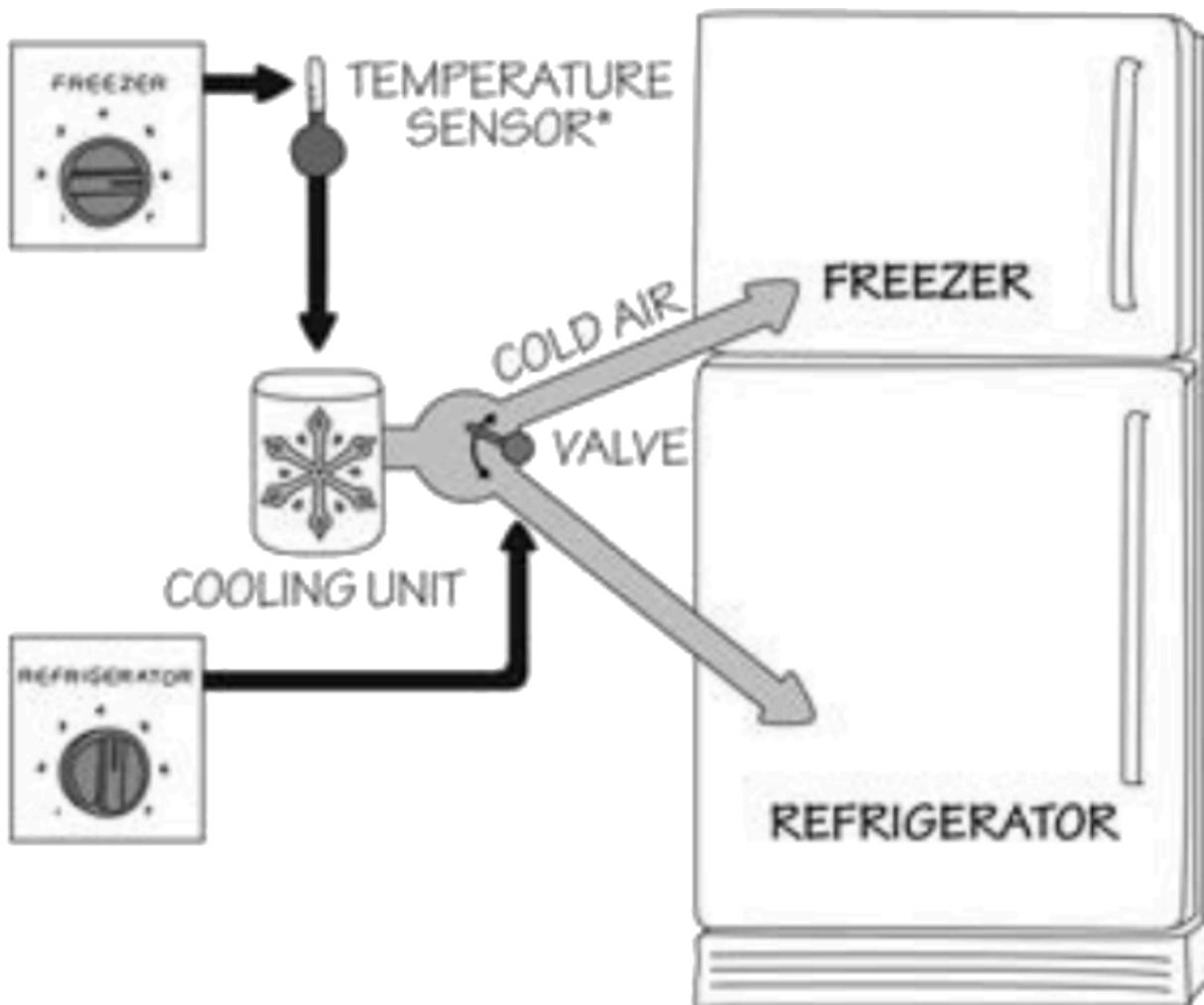
Modelo conceptual ↵ Modelo mental



system image



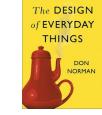
mental model



conceptual model

- ! Controladores não são independentes
 - 💡 “Freezer” controla o termostato
- ! “Refrigerator” controla % de ar frio entre compartimentos
- ! Esperar 24h por feedback

Modelo conceptual ↵ Modelo mental

- **Conceptual model** = como o designer quer que o utilizador veja o sistema
- No caso de uma má correspondência:
 - “When the designers fail to provide a **conceptual model**, we will be forced to make up our own **[mental model]**, and the ones we make up are apt to be wrong” (Don Norman 
 - “What users believe they know about a UI strongly impacts how they use it. Mismatched **mental models** are common, especially with designs that try something new.” (Nielsen Norman Group)
 - Erros de utilização, frustração, menor produtividade, ...



Modelo conceptual ↵ Modelo mental

- “Control of the refrigerator is made difficult because the manufacturer provides a **false conceptual model**. [...] Given **the correct model**, life would be much easier.” (Don Norman)



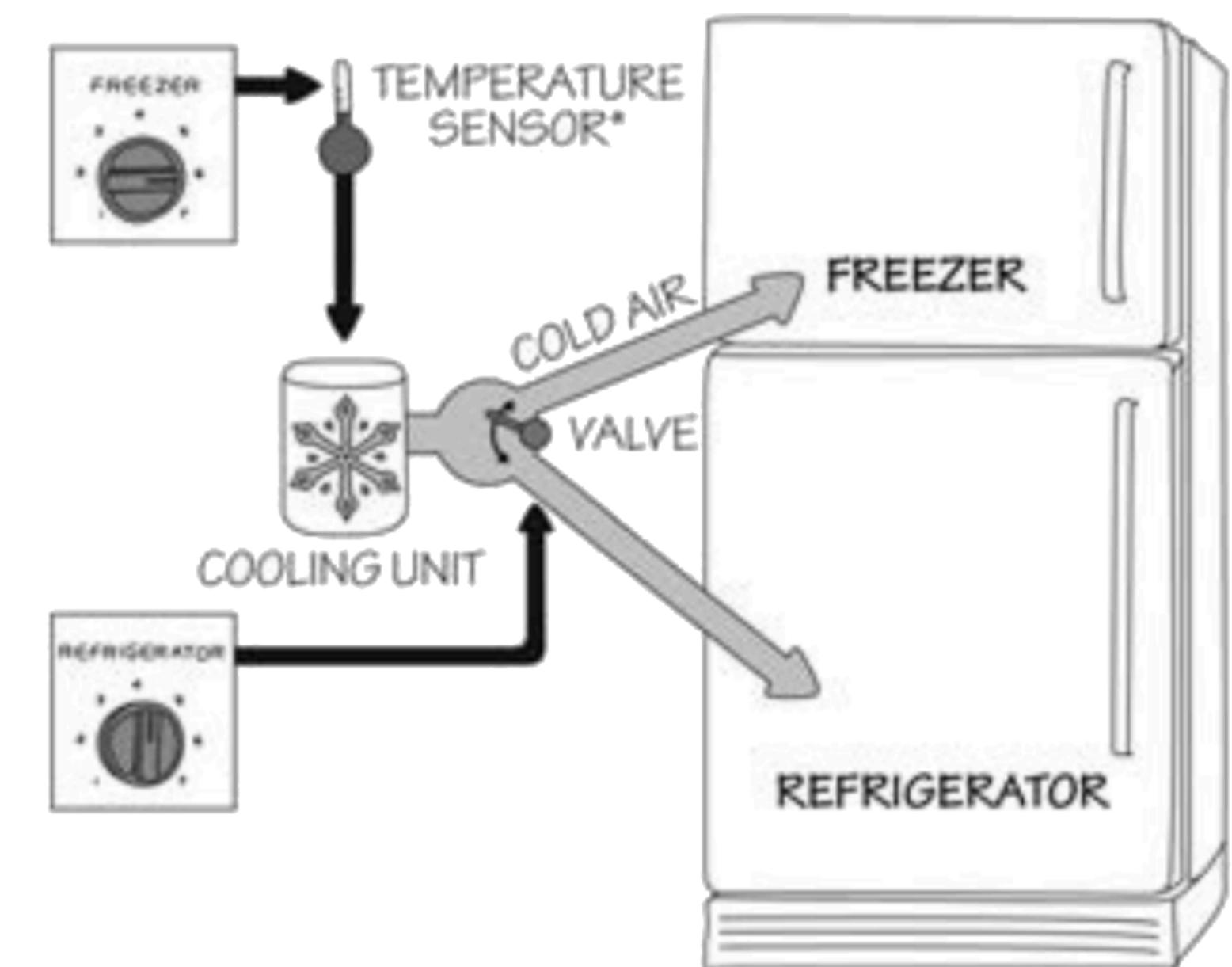
💡 Podemos tornar o modelo conceptual mais claro, e.g.:

- “Freezer” = “overall coldness”
- “Refrigerator” = “freezer/refrigerator ratio”

? É um bom modelo conceptual?

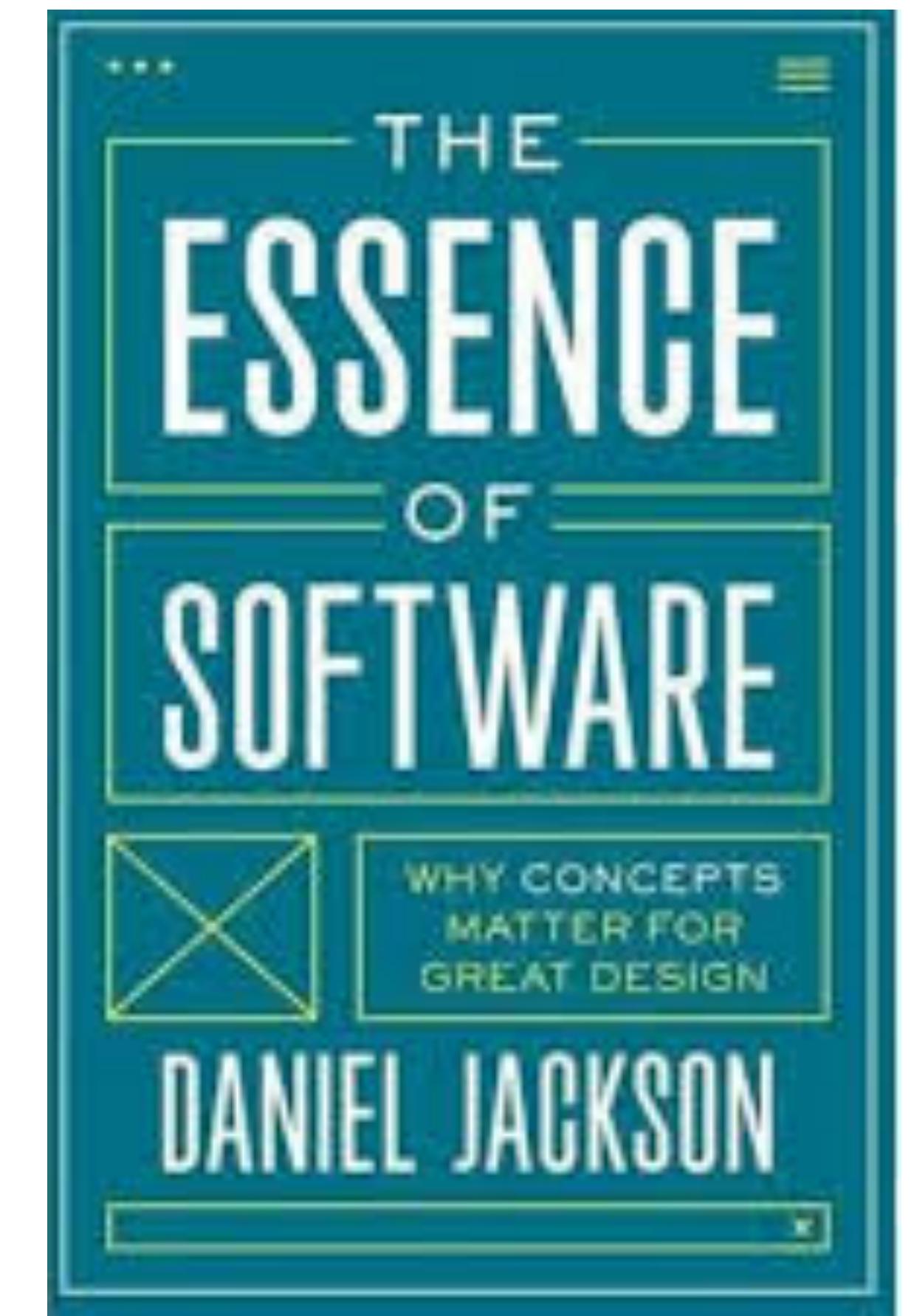
! Nem por isso. Podemos melhorar?

! Depende da realidade



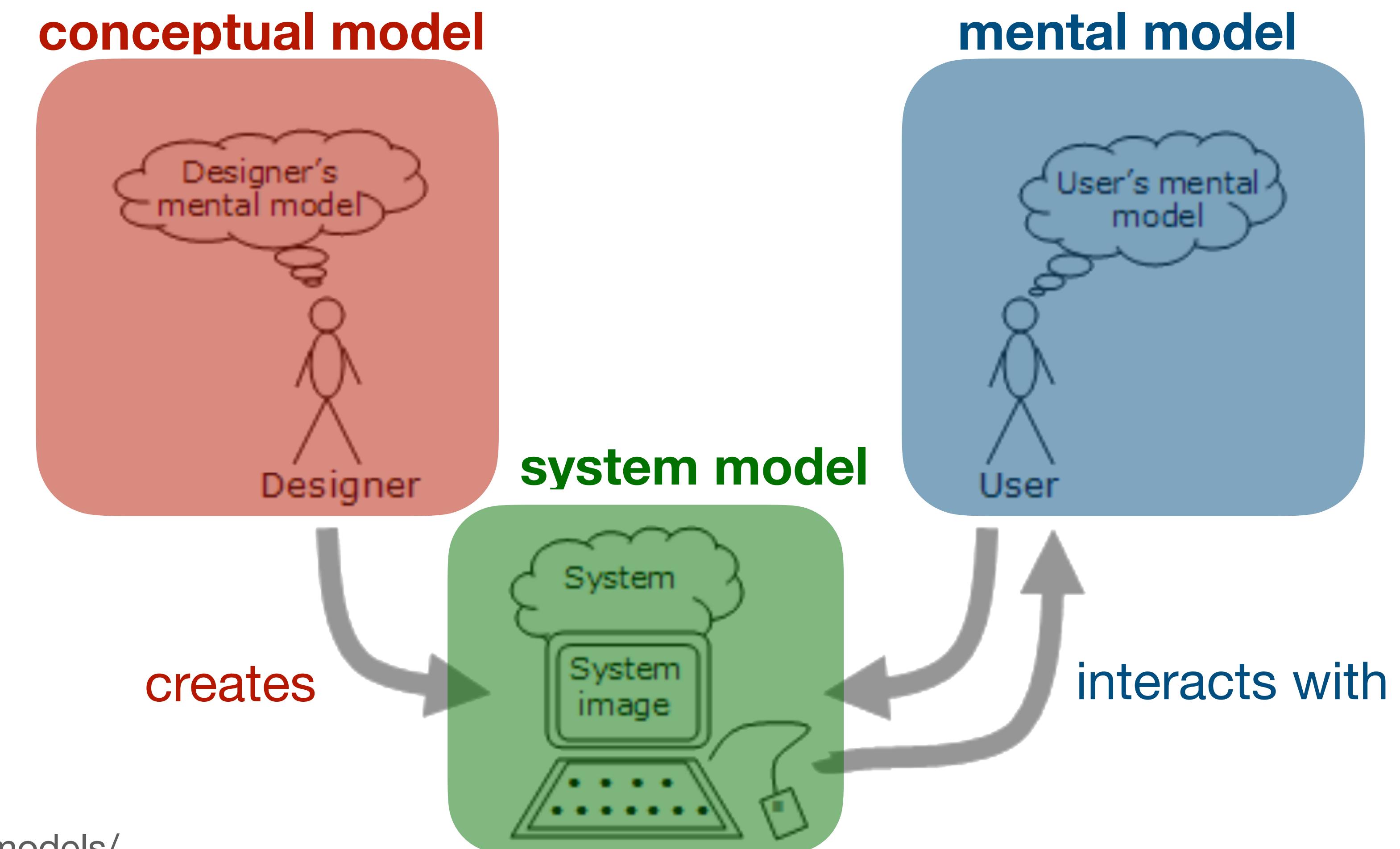
Modelo conceptual ↵ Modelo mental

- “The design would still be bad even if the controls were accurately labeled”
- “The **user’s purposes**— adjusting the temperatures of the fresh and frozen compartments—exist independently of any mental intention, and it is the fundamental misalignment of **purposes** and **concepts** that makes the design bad”
- “The notion of gulfs is invaluable for exposing usability snags, but in discovering a gulf between a **user intention** and a **system action**, one should not regard either the intention or the action as given, and **the best design solution may involve changing both**”



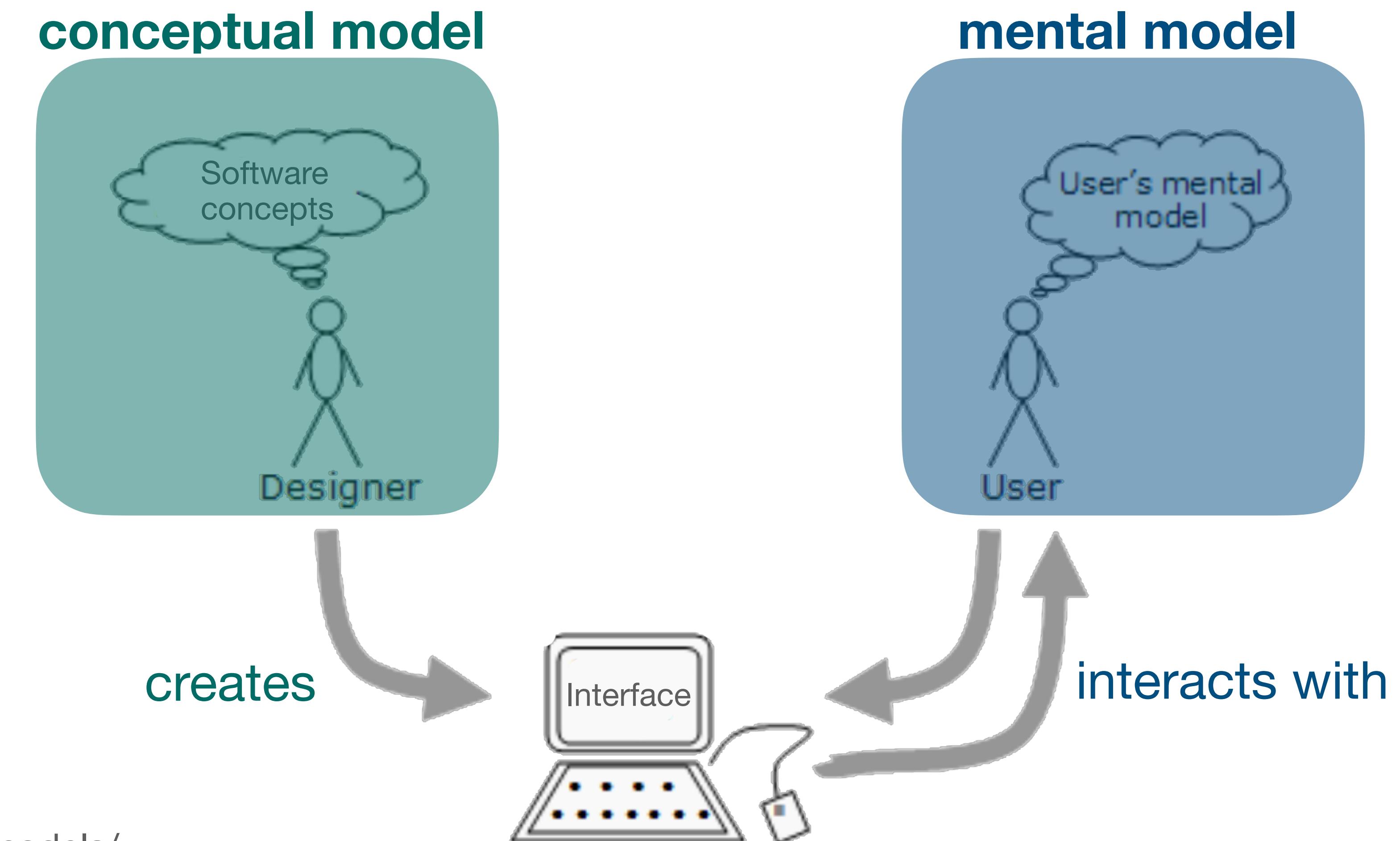
Software concepts

- Em software, temos a capacidade de “construir e moldar a realidade”
- “To speak of a **software** system [...] as having a **conceptual model** is misleading. It suggests there is some model distinct from the reality of the software itself. On the contrary, the **conceptual design** is the **design of the software**.”

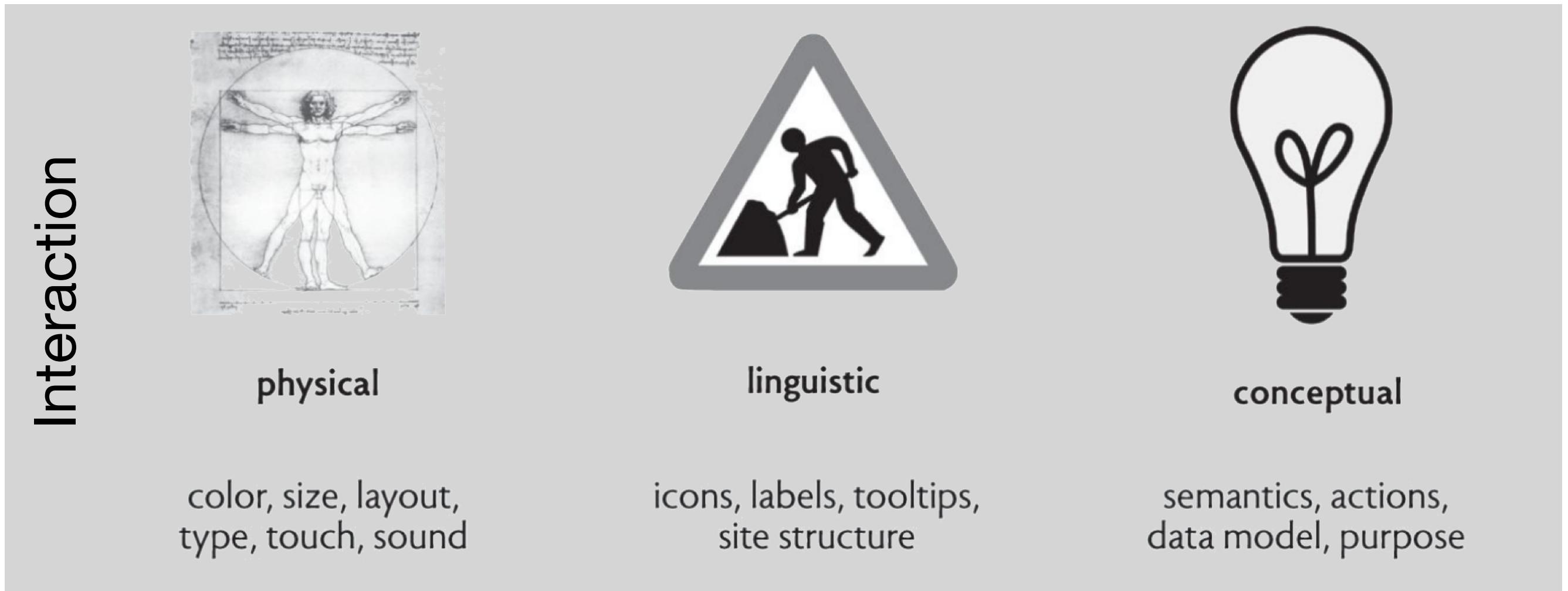
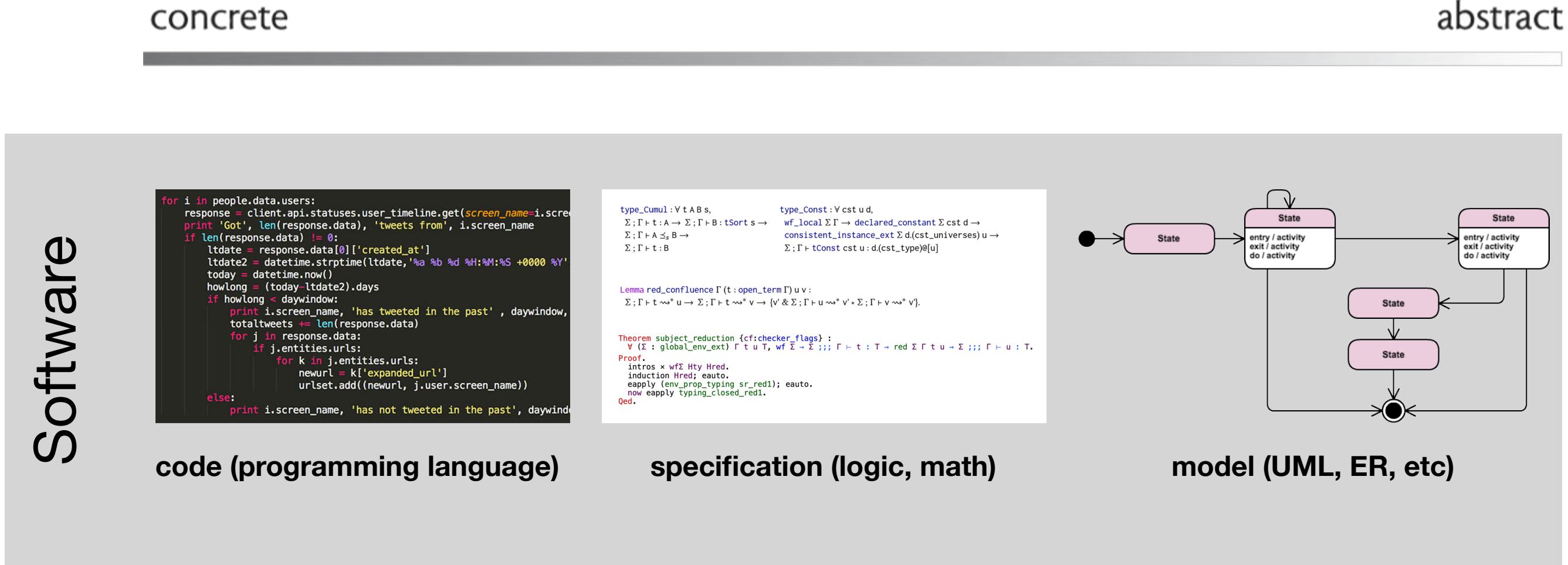


Software concepts

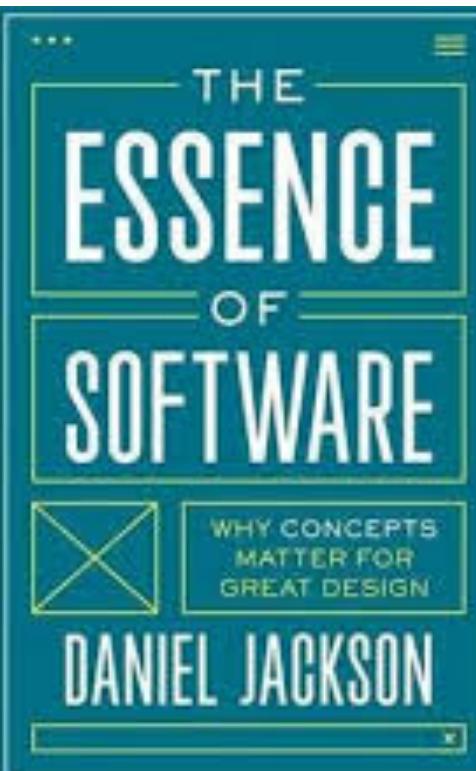
- Em software, temos a capacidade de “construir e moldar a realidade”
- “To speak of a **software** system [...] as having a **conceptual model** is misleading. It suggests there is some model distinct from the reality of the software itself. On the contrary, the **conceptual design** is the **design of the software**.”



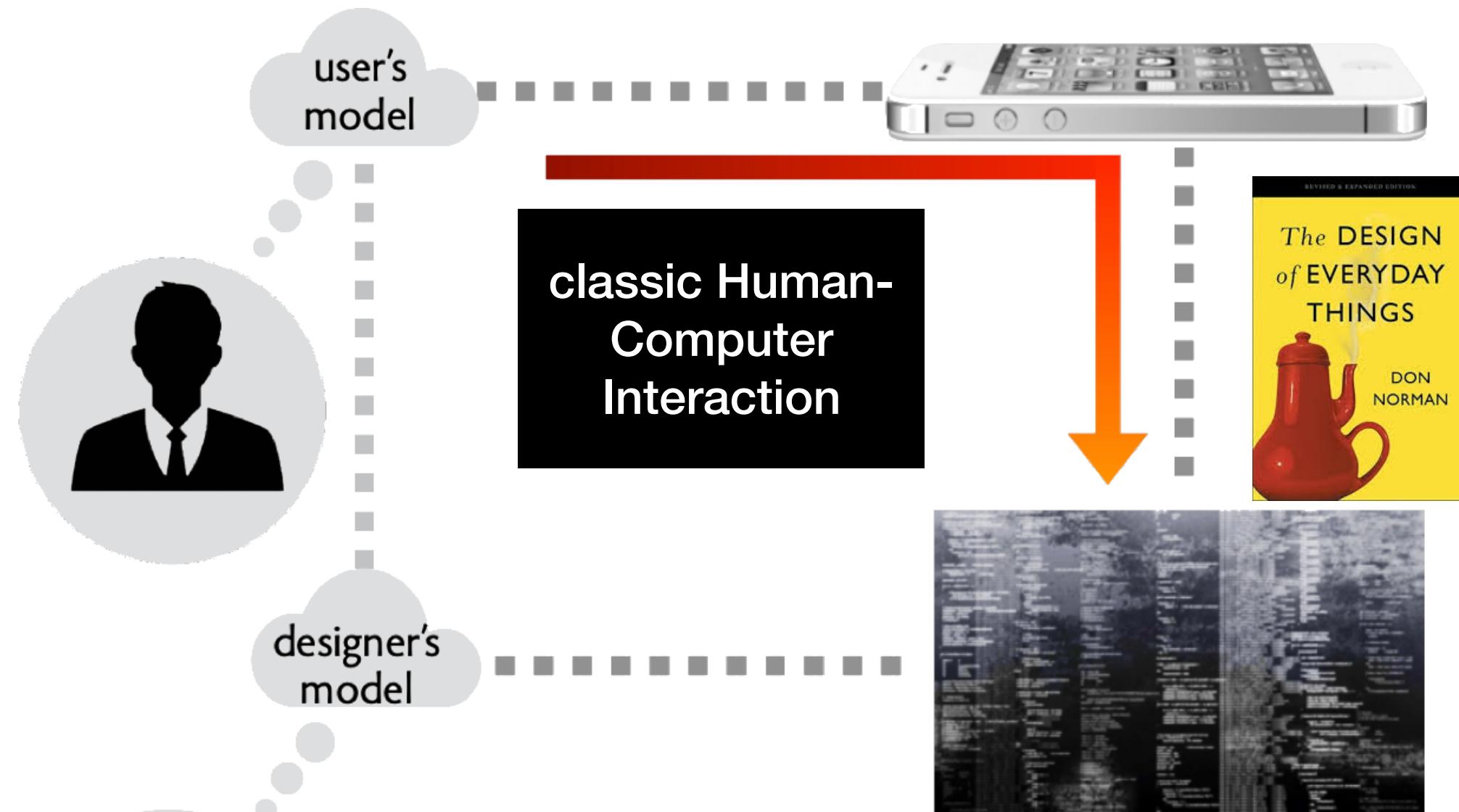
Software concepts: Representação / Abstração



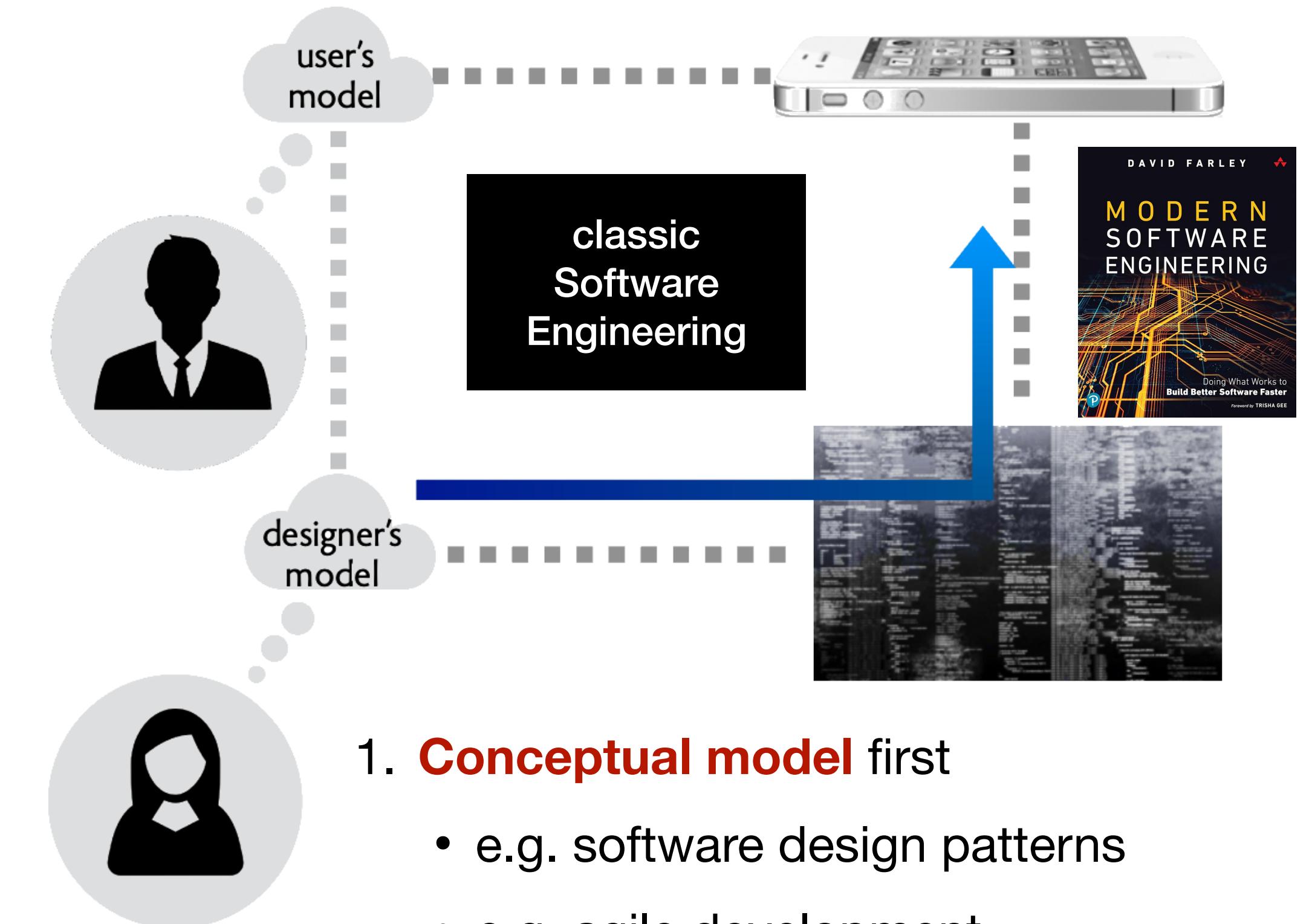
- Designers shall be able to “express **conceptual design** ideas without representing them in a concrete **user interface**”
- “Just as a single programming abstraction can have different representations, so a **concept** can be realized in different **user interfaces**”
- “Just as programmers think first about the abstraction and only second about the representation, so designers think at the conceptual level before the lower levels.”



Conceptual modeling

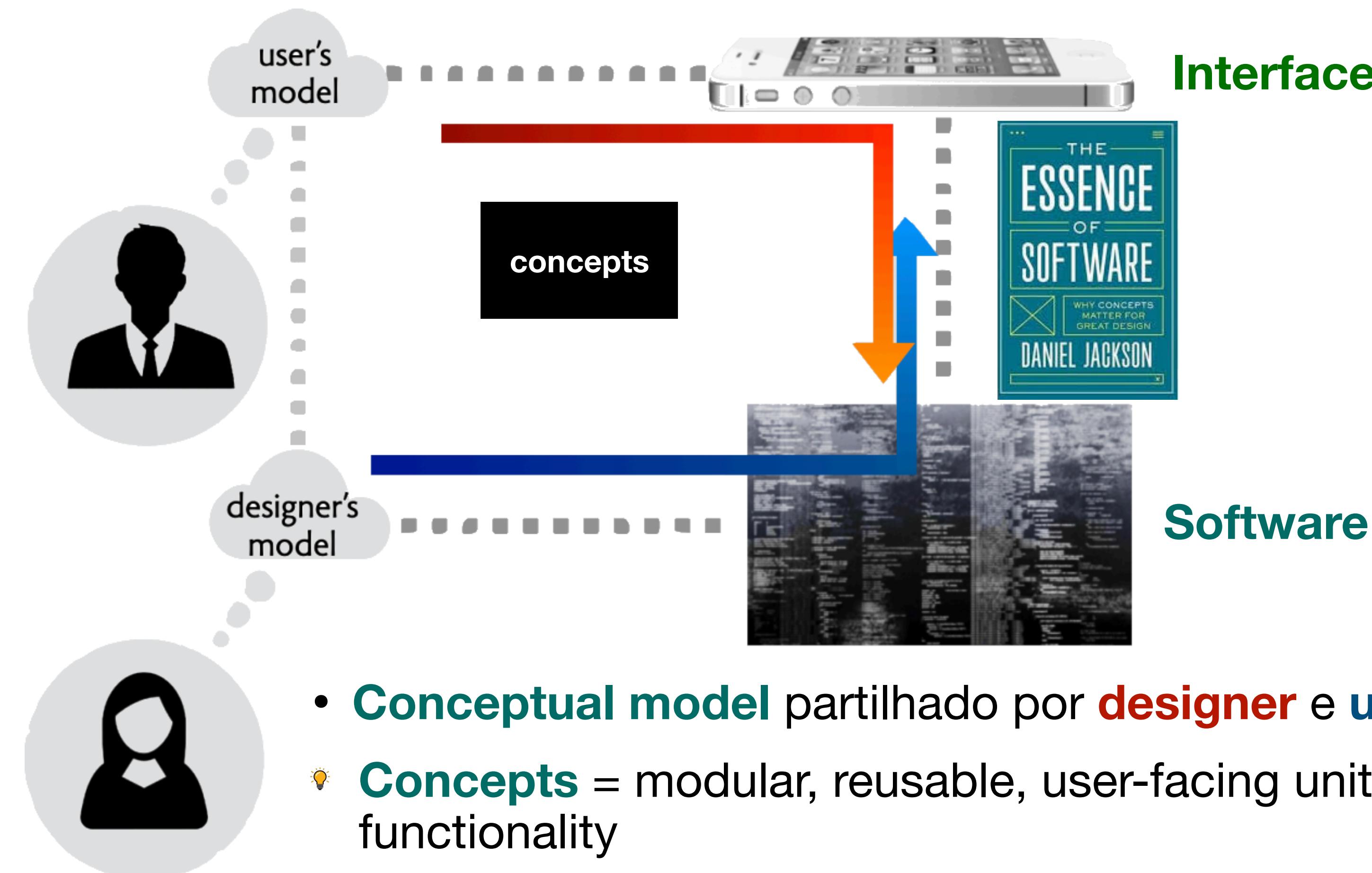


1. **Mental model** first
 - e.g. prototipagem
 - e.g. user studies
2. Design de um **conceptual model** que ligue a realidade do **sistema** à intuição do utilizador



1. **Conceptual model** first
 - e.g. software design patterns
 - e.g. agile development
2. Construção de um **mental model** que reflete o comportamento do **sistema** para o utilizador

Conceptual modeling



Um concept é...

- ...um incremento de funcionalidade...
 - independente de outros **concepts**
- ... que serve um **purpose**...
 - contribuindo para o **purpose** geral do **sistema**
- ...com o seu próprio **state**...
 - visível para o **utilizador**
- ...com as suas **actions**.
 - executadas pelo **utilizador**

Concepts: Trash

- “If you can find the trash can, you can run a computer” (Apple Lisa 1982)
- Mais do que uma metáfora
 - Tem um **purpose** definido
 - inclui **state** e **actions**
 - Intuição descrita num **operational principle**
 - “if you delete a file, it moves to a special folder; you can restore from there, but emptying it removes contents for good”



concept trash [Item]

purpose

to allow undoing of deletions

state

accessible, trashed: set Item

actions

create (x: Item)

when x not in accessible or trashed

add x to accessible

delete (x: Item)

when x in accessible but not trashed

move x from accessible to trashed

restore (x: Item)

when x in trashed

move x from trashed to accessible

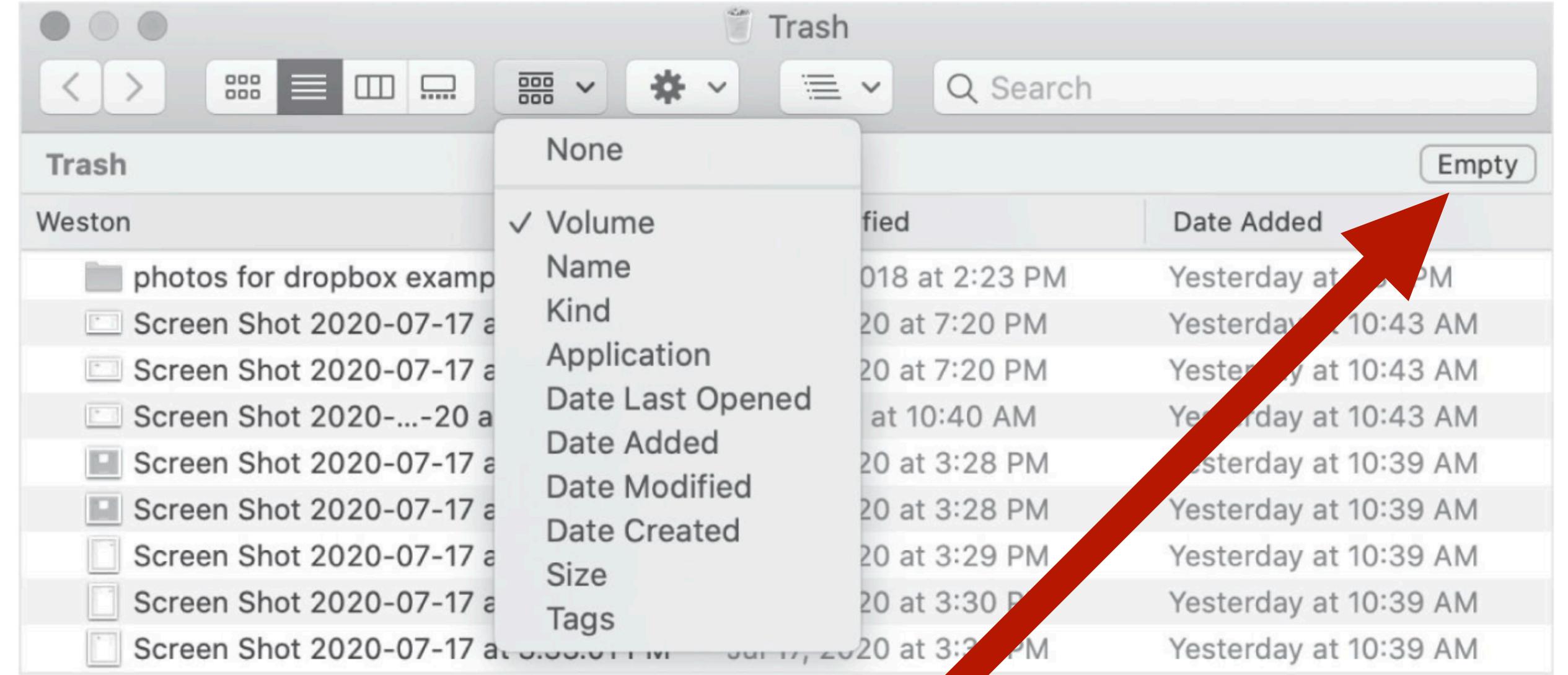
empty ()

when some item in trashed

remove every item from trashed

Concepts: Trash

- Composição de concepts
 - **Trash visto como uma folder**
 - Items no **trash** são **folders**
 - “Date Added” no **trash** = data em que o item foi removido
- ? Questões
 - Um só **trash**?
 - O que acontece com devices externos?



Sincronização de concepts

- Esvaziar **trash**?
- Mover ficheiros de/para **trash** coerente com mover entre **folders**
- Apagar na **trash folder** = repôr

Exemplo: Dropbox

- Ava (AA) partilhou uma pasta chamada “Bella Party” com Bella (BB)
 - Incluindo uma sub-pasta “Bella Plan”

The screenshot shows the Dropbox interface with a search bar at the top left and user icons (AA and BB) at the top right. Below the search bar, the text "Dropbox > Bella Party" is displayed. An "Overview" section follows, featuring a header with "Name ↑", "Members", and a sorting dropdown. A table lists one item: "Bella Plan" (with a star icon), which has "2 members". There is also a "..." button next to the item.

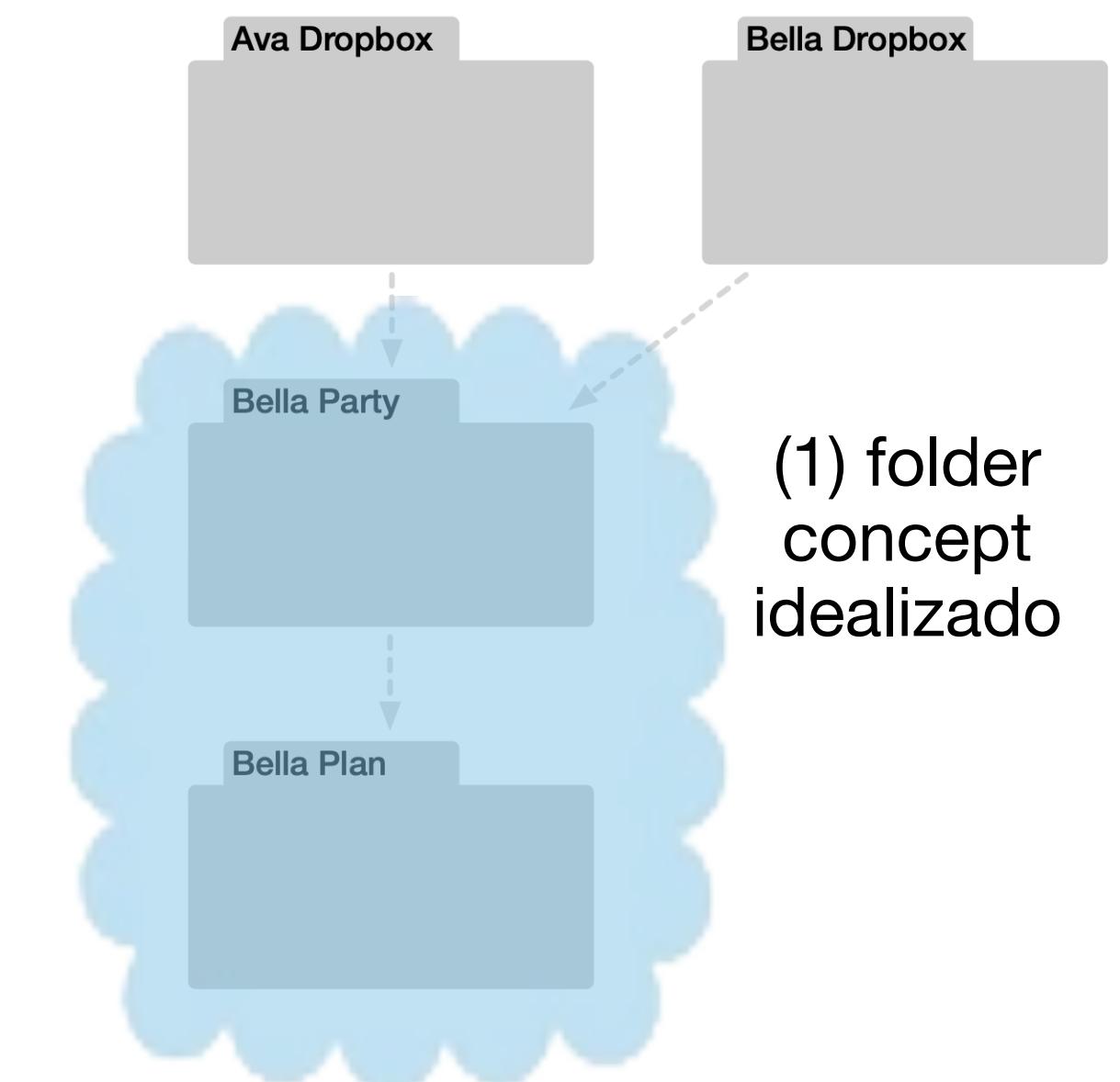
The screenshot shows the Dropbox interface with a search bar at the top left and user icons (AA and BB) at the top right. Below the search bar, the text "Dropbox" and "Overview" are displayed. An "Actions" section follows, with a header for "Name ↑", "Members", and a sorting dropdown. A table lists one item: "Bella Party" (with a star icon), which has "2 members". There is also a "..." button next to the item.

The screenshot shows the Dropbox interface with a search bar at the top left and user icons (AA and BB) at the top right. Below the search bar, the text "Dropbox" and "Overview" are displayed. An "Actions" section follows, with a header for "Name ↑", "Members", and a sorting dropdown. A table lists one item: "Bella Party" (with a star icon), which has "2 members". There is also a "..." button next to the item.

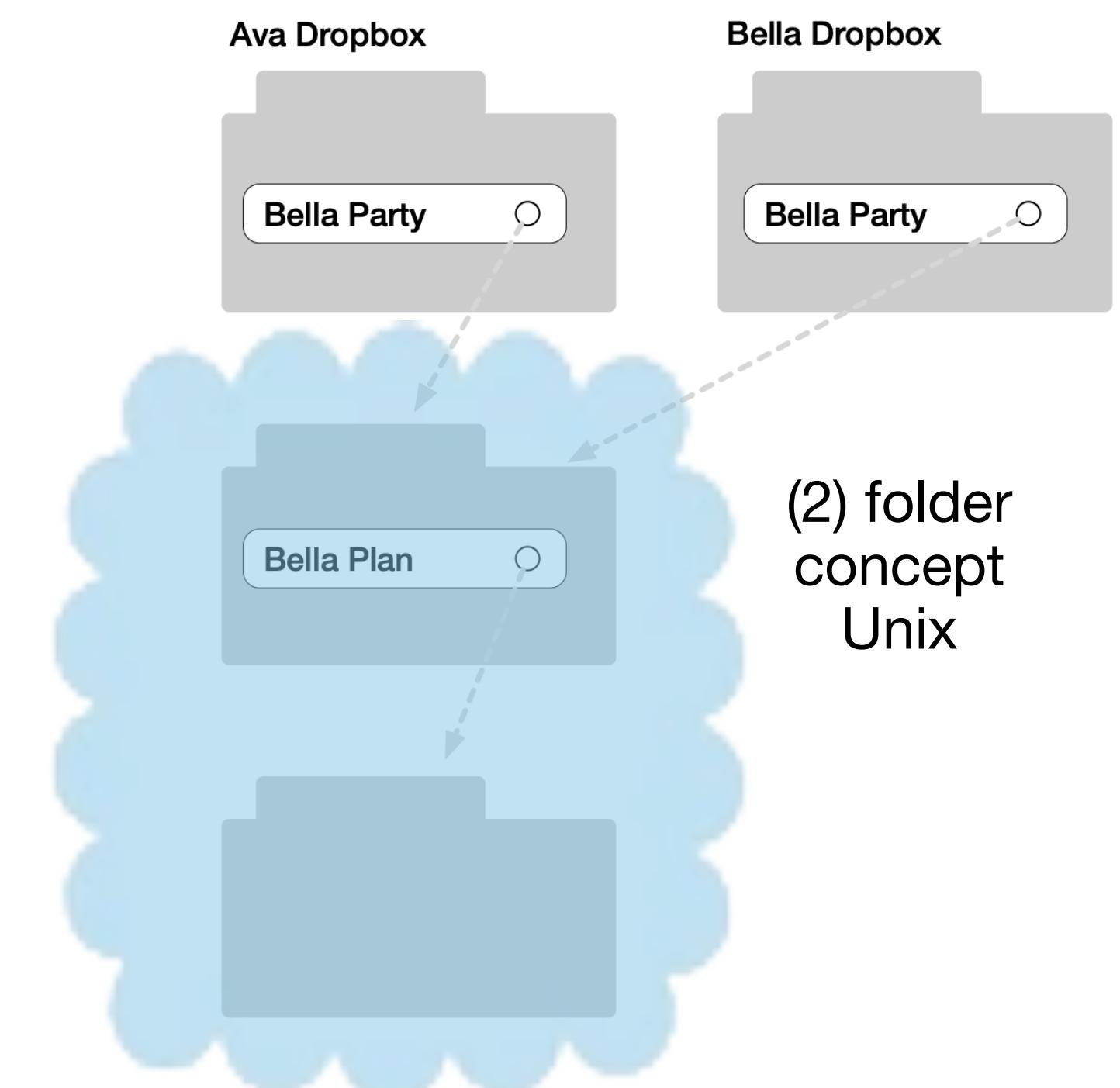
Exemplo: Dropbox

? Qual o **modelo conceptual**?

- Se Bella renomear “Bella Party” para “My Party”
 - Renomeia só a sua pasta, Ava vê nome original
- Se Bella apagar a pasta “Bella Party”
 - (1) Apaga para ambas (2) Apaga só para si
- Se Ava apagar a pasta “Bella Plan”
 - Apaga para ambas



(1) folder
concept
idealizado



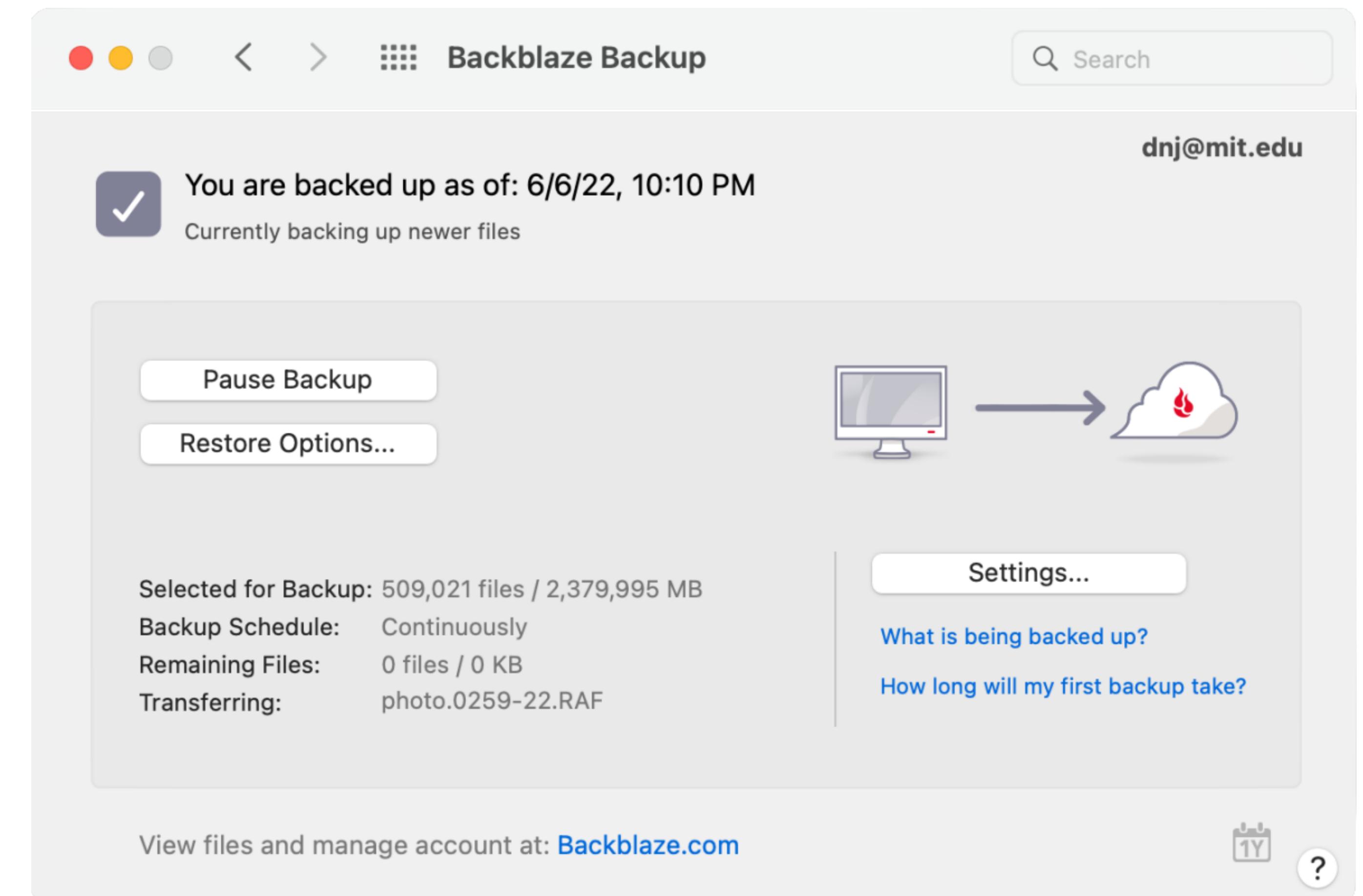
(2) folder
concept
Unix

Exemplo: Backblaze

- Considerem uma aplicação de backups automáticos para a cloud

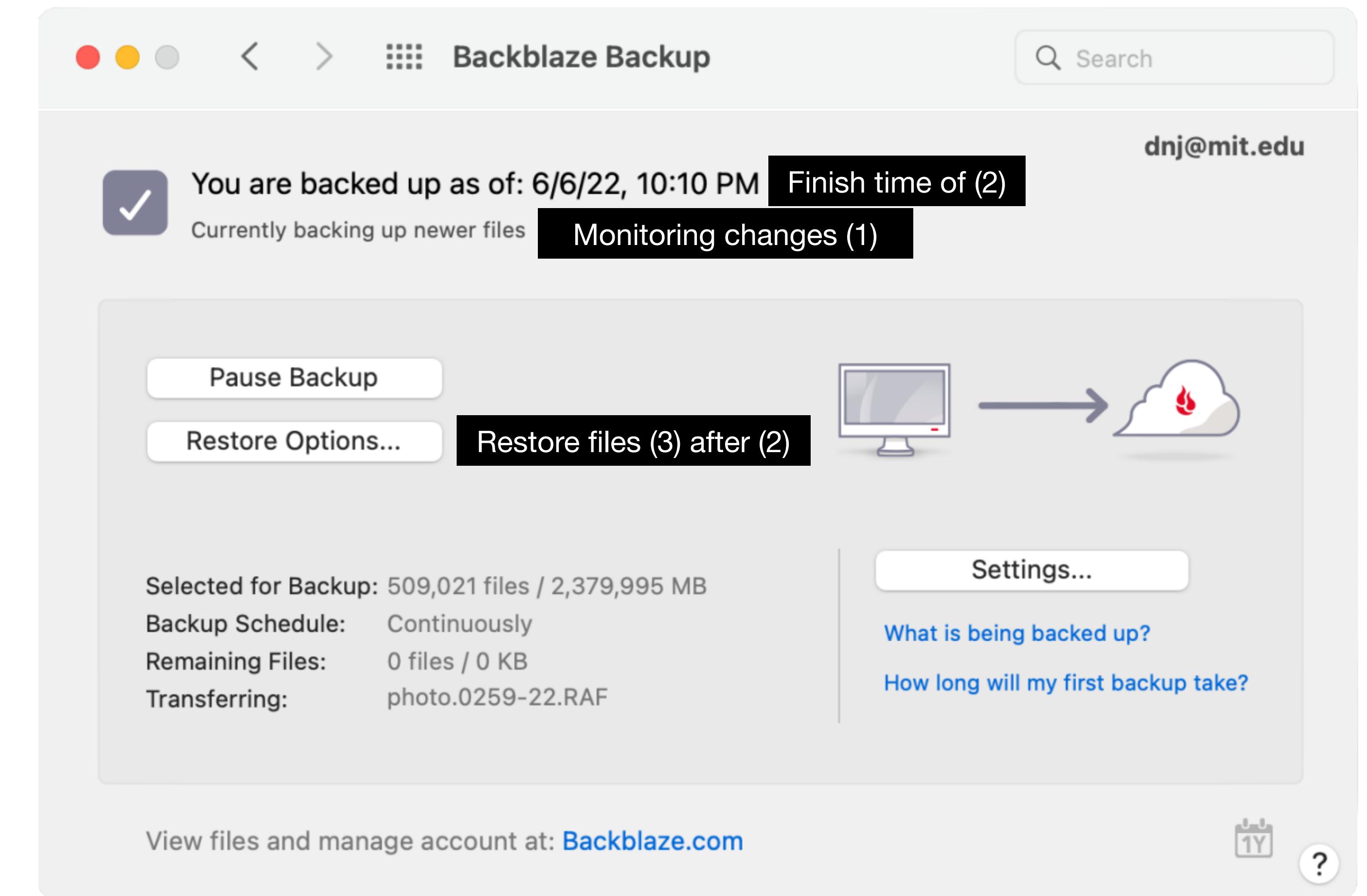
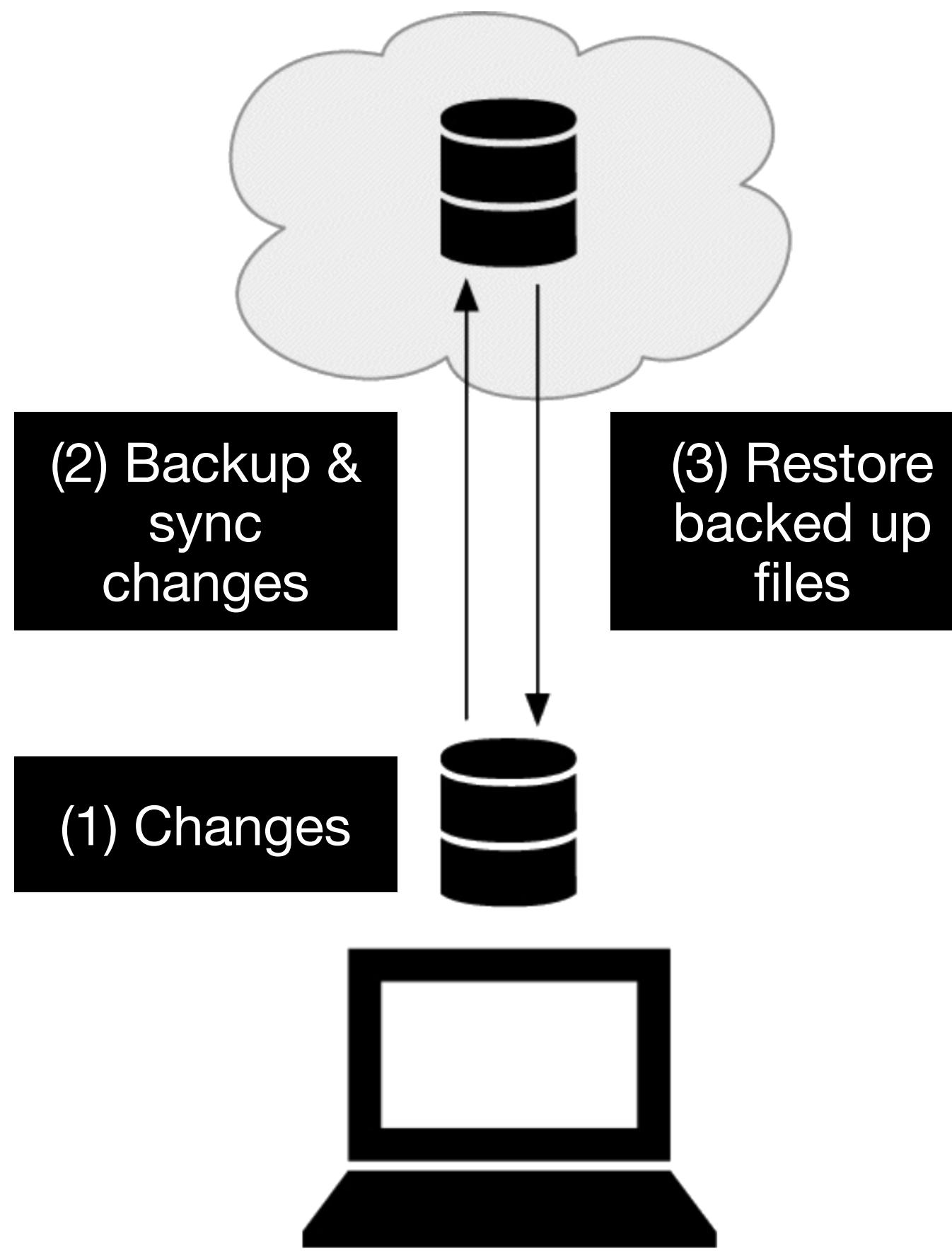
? Qual o **modelo conceptual**?

- Está atualmente a fazer backup?
- Que modificações antes das 10:10PM estão guardadas? Onde?



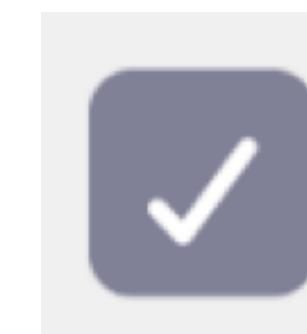
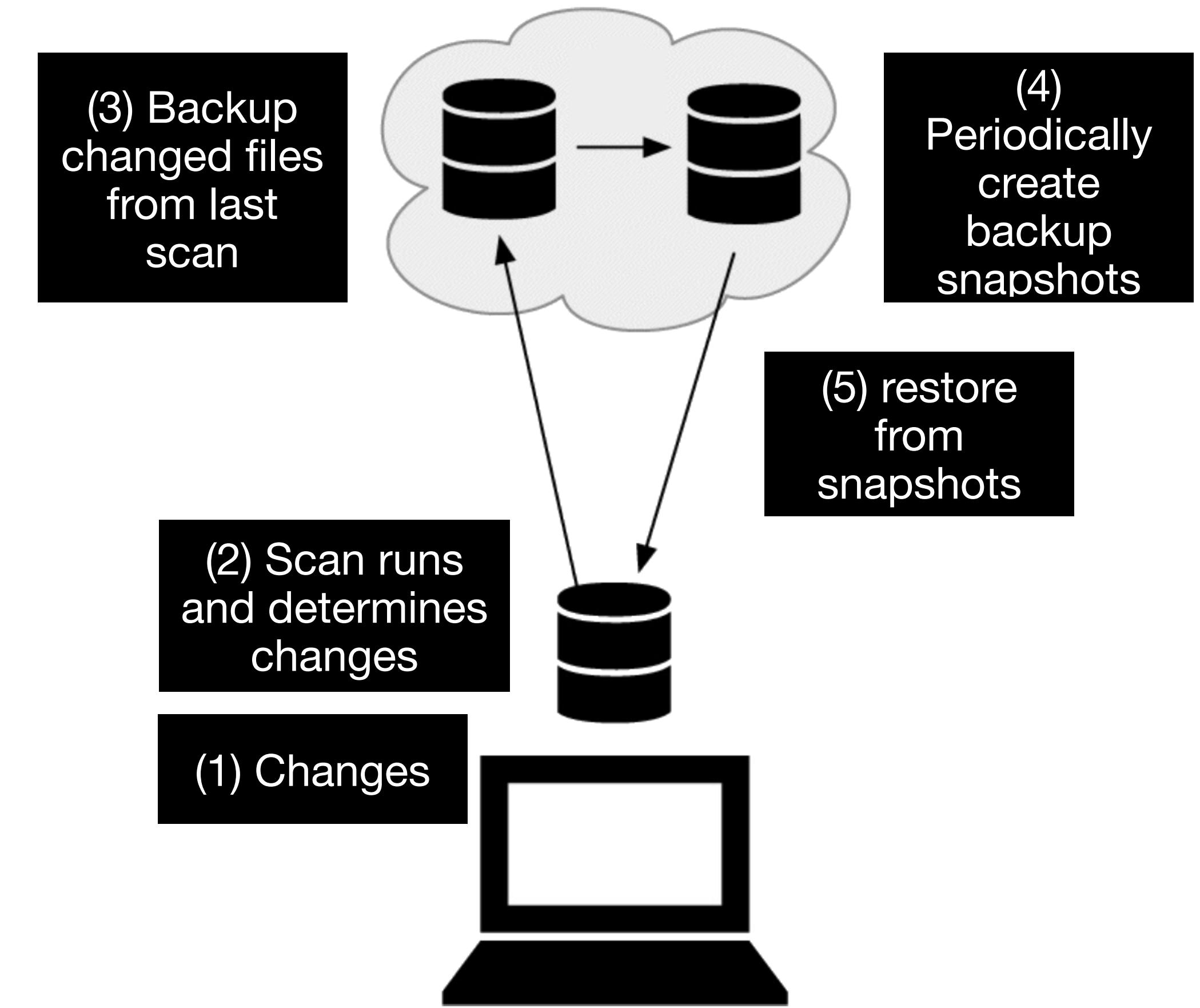
Exemplo: Backblaze

- Um possível modelo



Exemplo: Backblaze

- O modelo “real”
 - Modificações não são diretamente sincronizadas
 - Scan → Backup
 - Mais do que uma localização na cloud
 - Backups → Snapshots
 - Restore de snapshots



Last backup completed at: 6/6/22, 10:10PM.
This includes all files saved prior to the scan completed at:
6/6/22, 10:00PM.

Exemplo: Backblaze

concept FileStore [Name, Content]

purpose store files persistently

principle after creating and updating a file, you can get the content

state

a set of files
for each file
name, contents

actions

create (n: Name, c: Content)

update (n: Name, c: Content)

delete (n: Name)

get (n: Name): Content

concept Backup [Name, Content]

purpose retrieve old version of files

principle after a file's contents are saved, they can be retrieved later by date

state

a set of files with versions
for each file
name, contents, date

actions

save (n: Name, c: Content)

restore (n: Name, d: Date): Content

concept Workset [Item]

purpose process items in batches

principle after items are added, and processing is started, the items are processed

state

current set of items being worked on
next set of items to work on

actions

start ()

requires current == {}

current = next

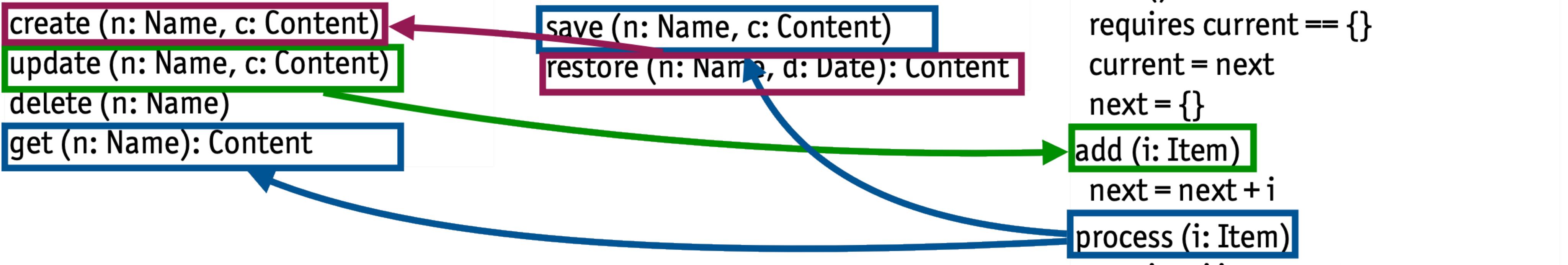
next = {}

add (i: Item)

next = next + i

process (i: Item)

requires i in current
current = current - i



Concepts: composição

💡 Concepts são independentes

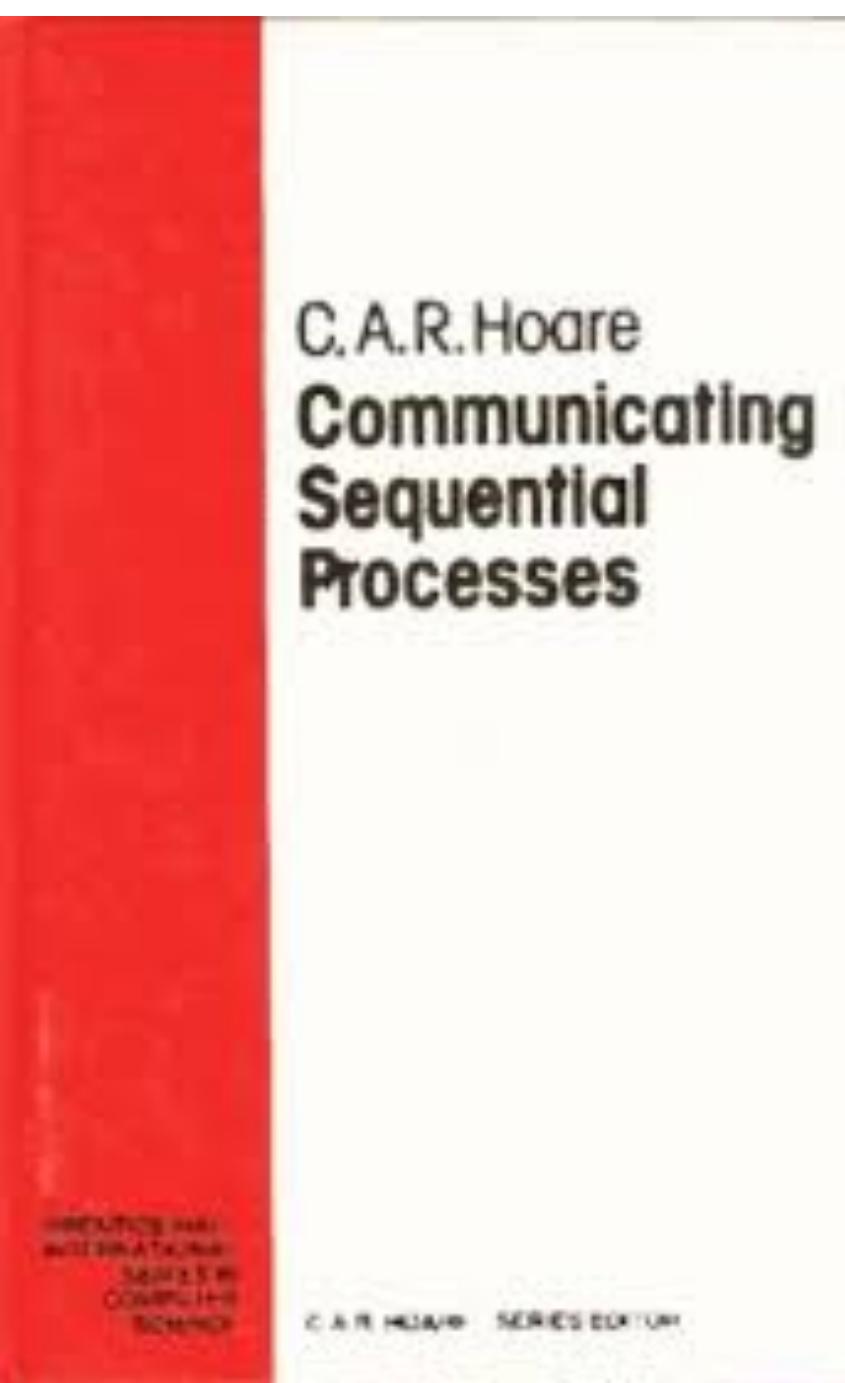
- Existem simultaneamente
- Têm estado próprio
- Executam ações em paralelo

💡 Composição através de sincronização

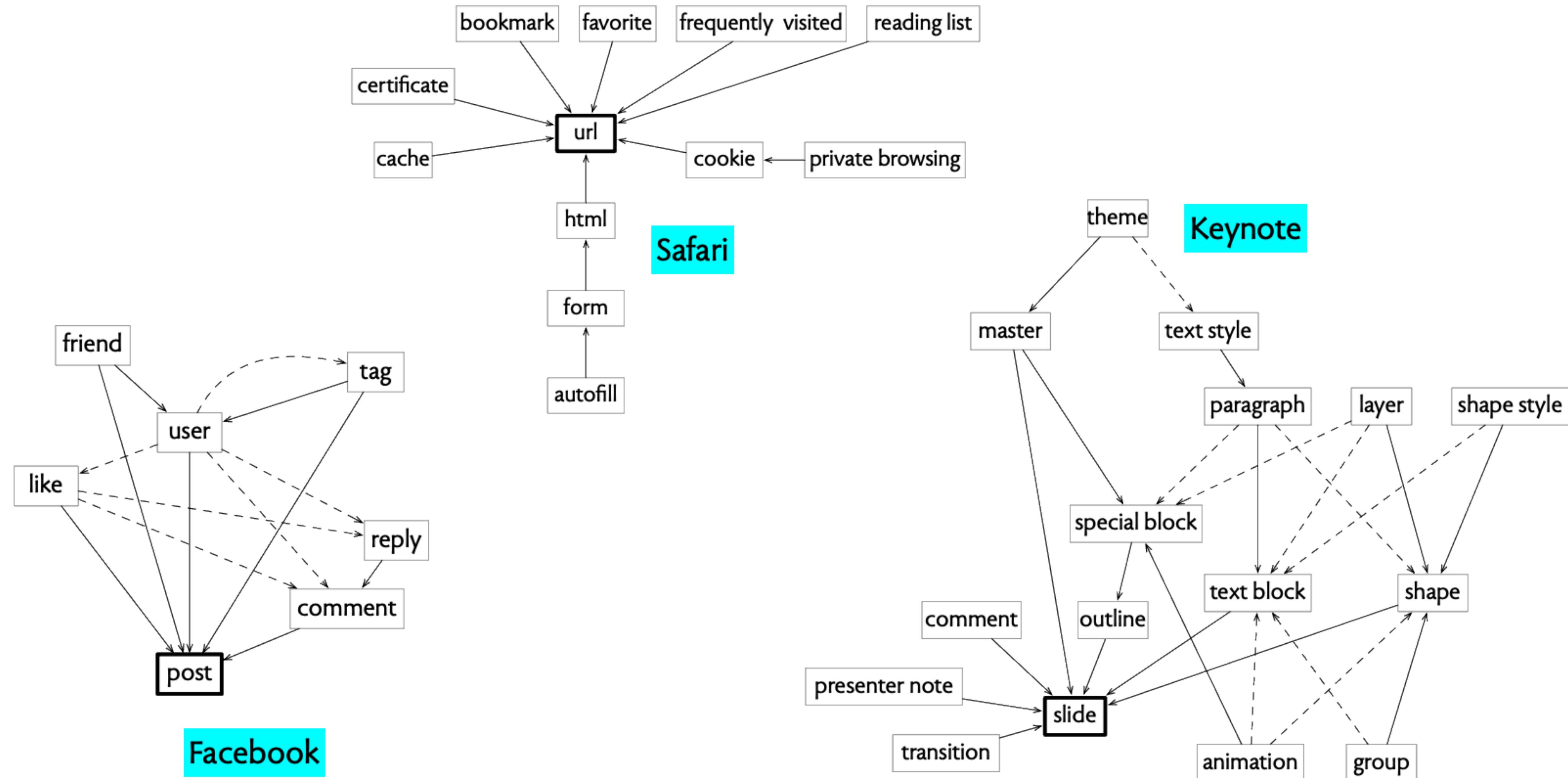
- A. action \Rightarrow B. action

❗ Não significa que as duas ações executam atomicamente

- Apenas que B. action tem que ocorrer no futuro se A. action ocorrer
- Outras ações podem ocorrer pelo meio



Concepts caracterizam apps



Concepts caracterizam apps

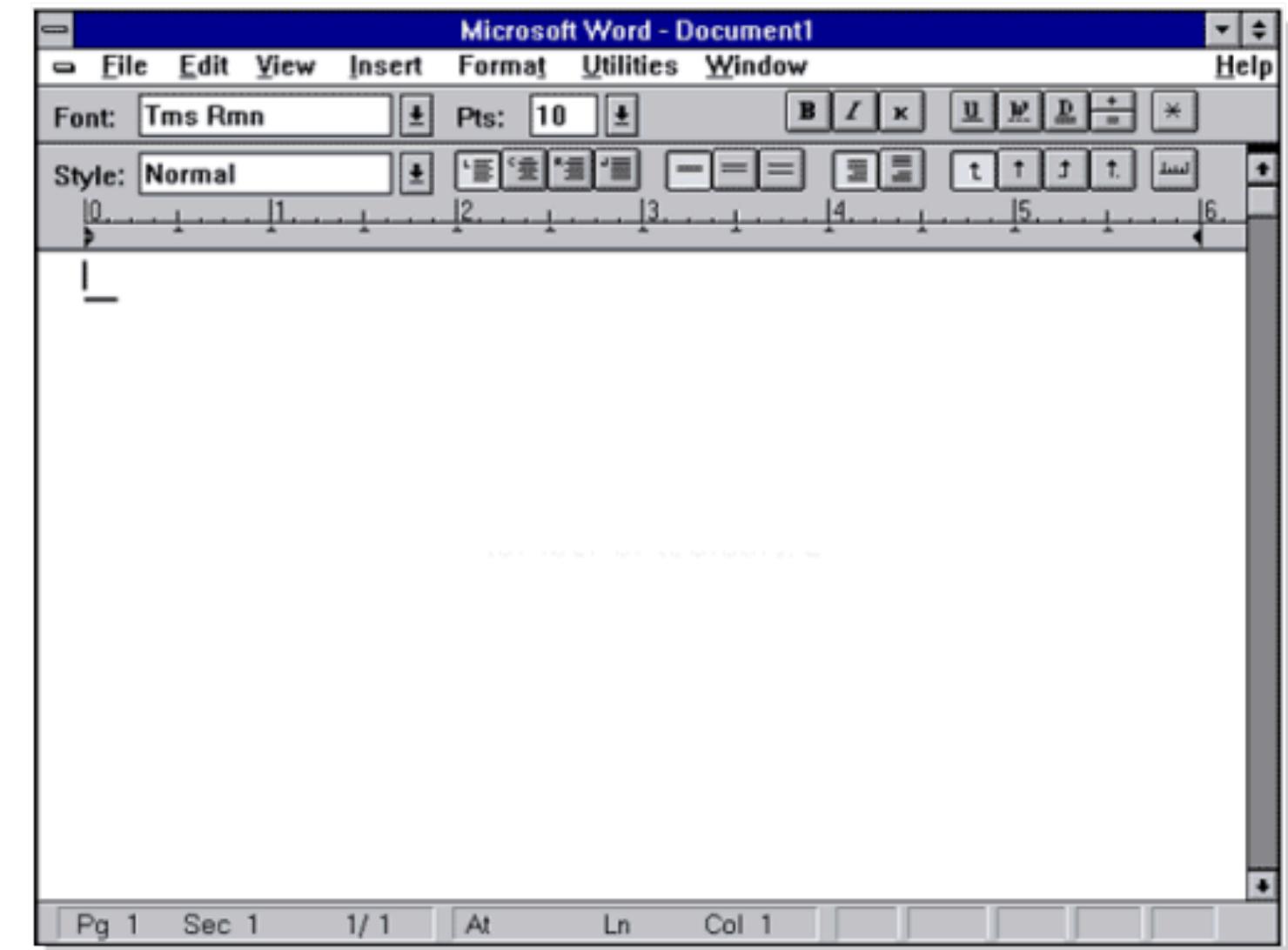
```
44 - approaches, not satisfying: surely something essential
45
46 Take a step back: concepts
47 - examples: Microsoft Word's line and para
48 - where concepts come from
49 - implications of concepts
50 - how concepts distinguish apps (Word from LaTeX) and app classes (Word from bold)
51 - what's software made off: concepts
52
53 Towards a theory of concepts:
54 - what's a concept?
55 state machine of their example
56 abstract state, usually structural, explains behavior
57 can explain with an operational principle without referring to purpose
58 more complex examples: style, alias, trash, tagging, selection or Po
59 in all cases the concept does not fulfill a function HELPFULLY
60
61 - granularity:
62 smallest concept that's a functionality increment
63 e.g. can have layer without mask
64 but may choose to group into larger concept
65
66 - dependence:
67 concepts have dependences based on functional subsets
68 what's the point of that?
69
70 - criteria: compelling, coherent, controllable (decoupled), general, complete, consistent,
```

text editor

line

character set

markup

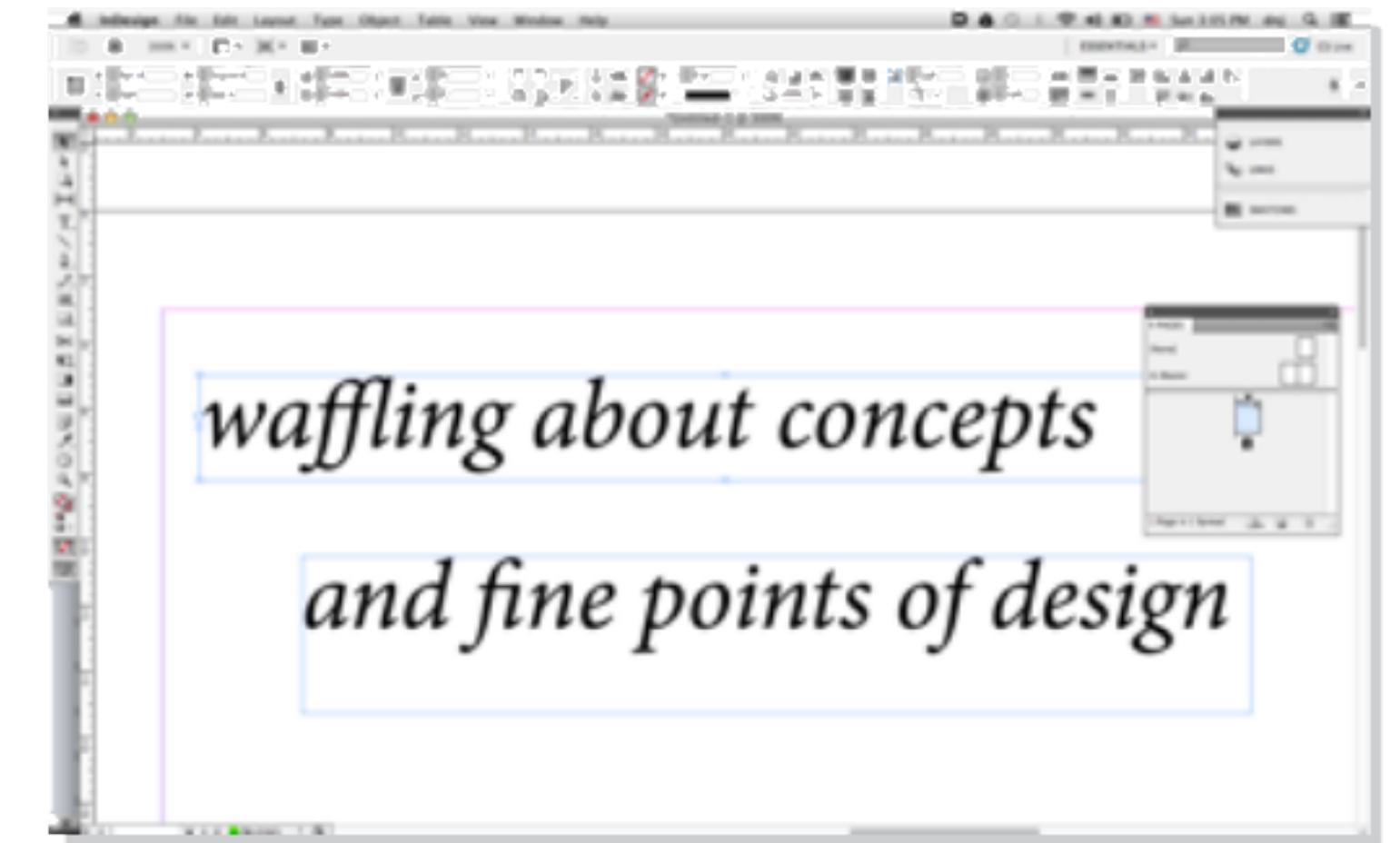


word processor

paragraph

format

style



desktop publishing app

paragraph

format

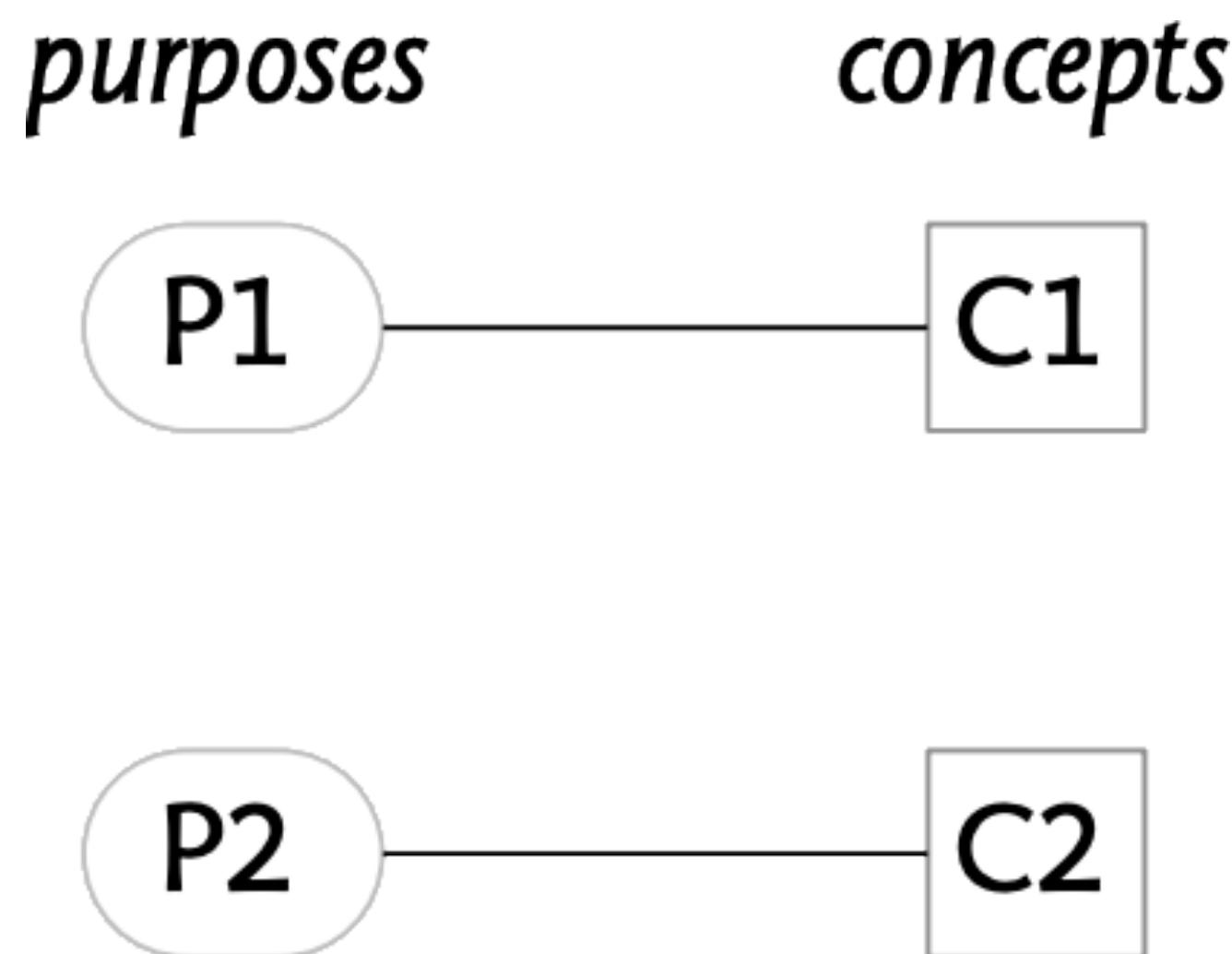
style

page

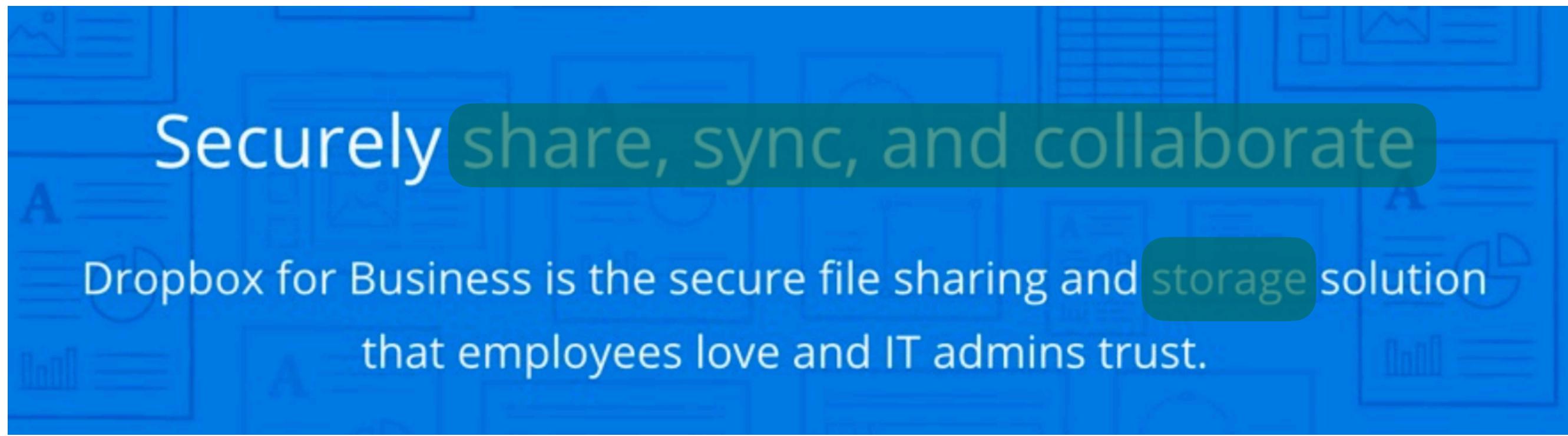
textflow

Concepts: Mapeamento

- Princípio fundamental
 - “Design is driven by purpose”
 - Cada **concept** deve ser motivado por exatamente um **purpose**
- Mapeamento ideal:



Exemplo: Dropbox



- **share**: control who can read your files
- **sync**: keep files on multiple machines consistent
- **collaborate**: support multi-user editing of documents
- **store**: expand space available by storing files in the cloud

P1

C1

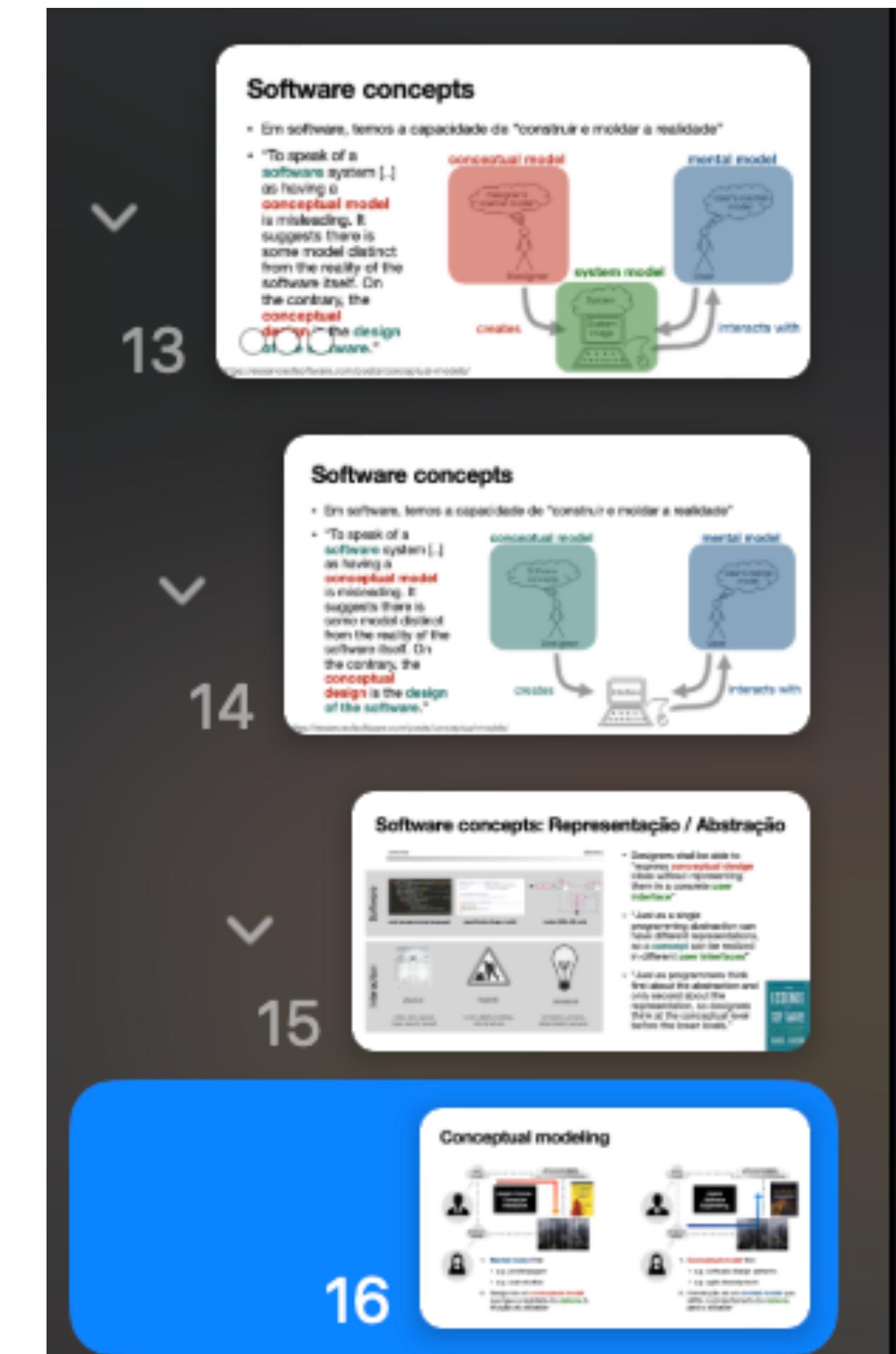
Concepts: Mapeamento

- Hierarquia de slides
- + KeyNote organiza slides numa árvore
- PowerPoint suporta só um nível

PowerPoint



KeyNote

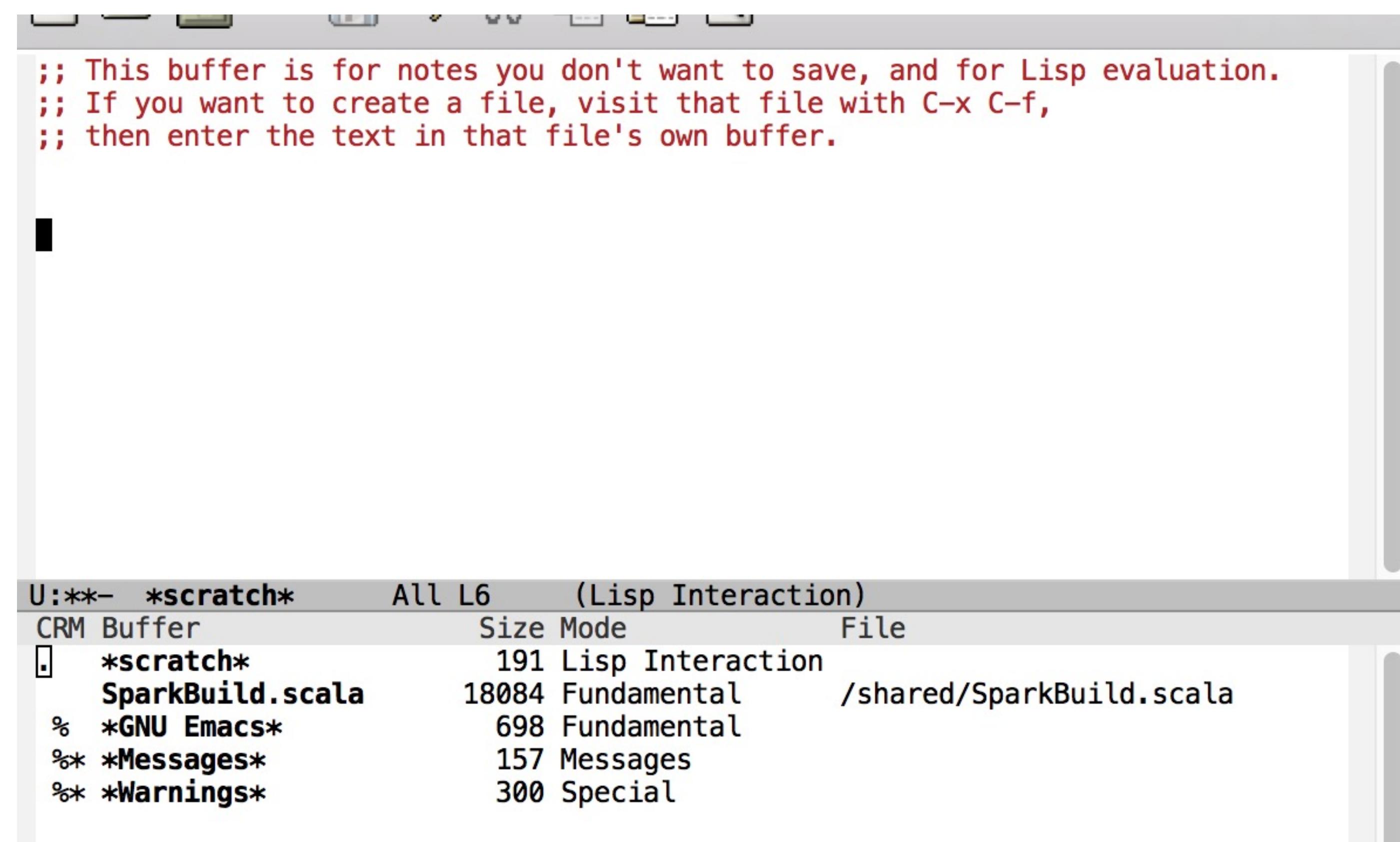


P2

Concepts: Mapeamento

- Qual o purpose do buffer concept?
 - Performance...
 - Uma preocupação do utilizador?
 - Não

Emacs



The screenshot shows an Emacs interface. The top part of the window displays the text:

```
;; This buffer is for notes you don't want to save, and for Lisp evaluation.
;; If you want to create a file, visit that file with C-x C-f,
;; then enter the text in that file's own buffer.
```

Below this, there is a small black square icon followed by a vertical line.

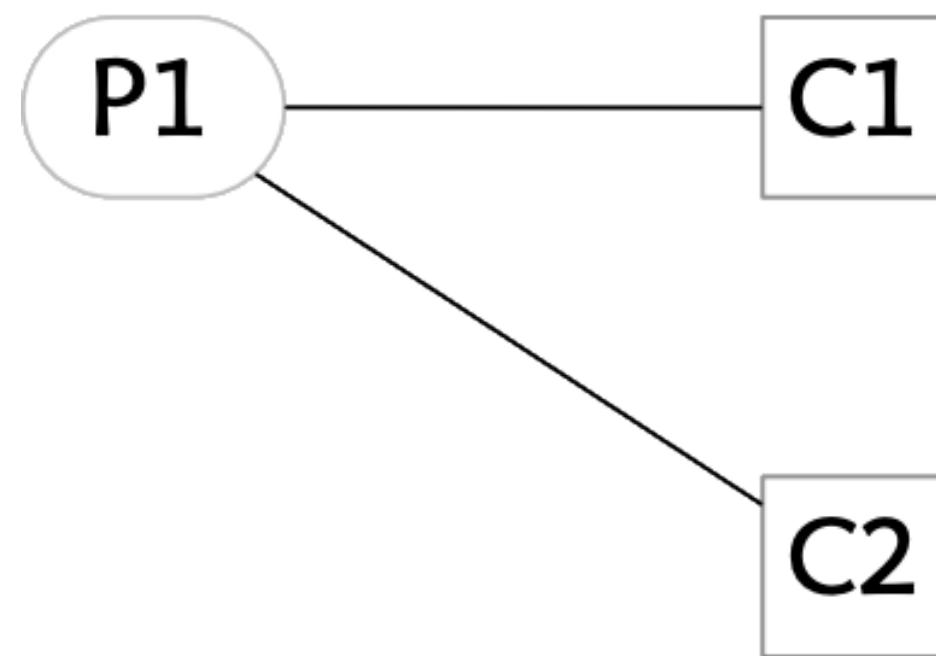
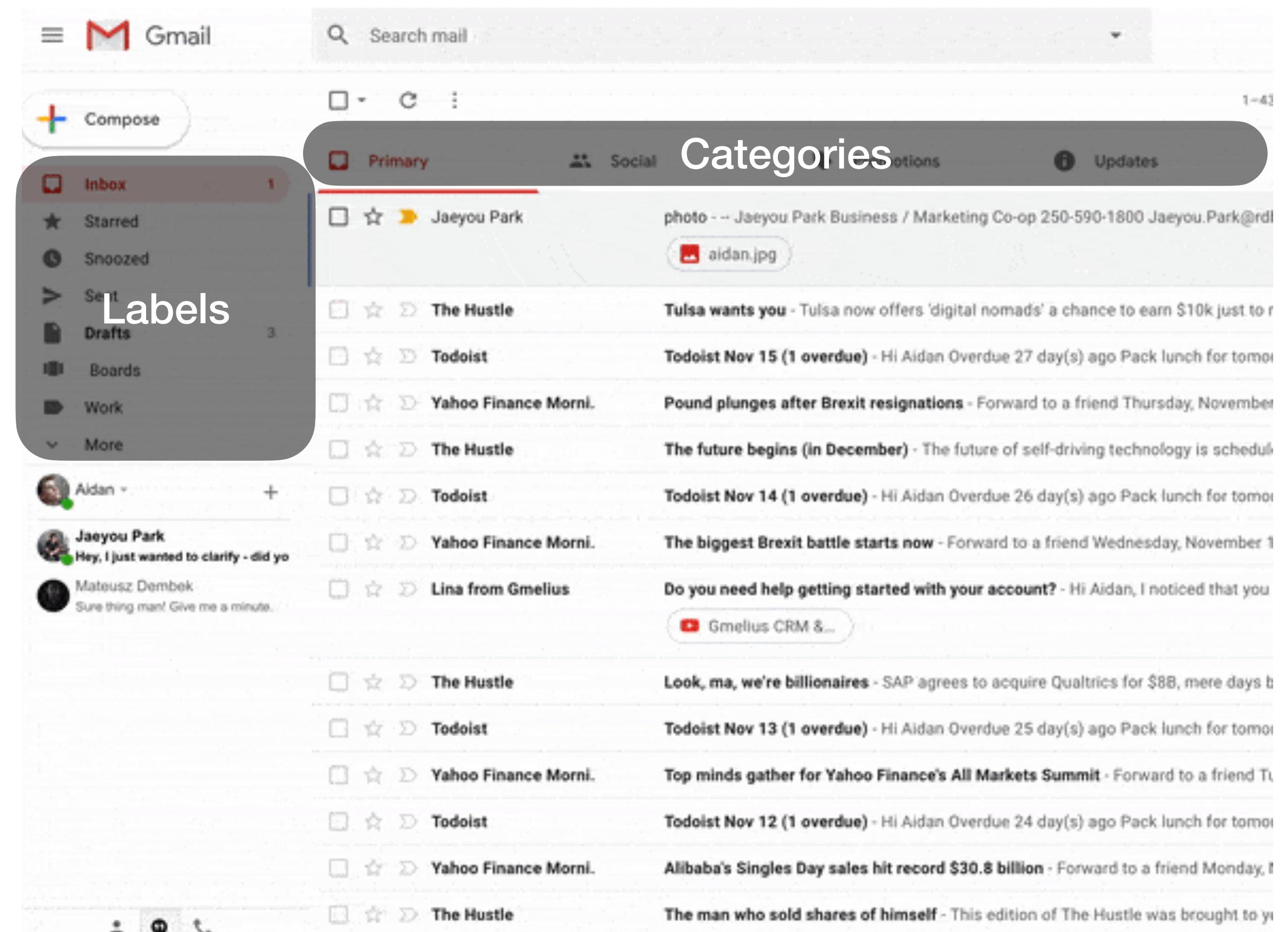
U:**- *scratch* All L6 (Lisp Interaction)			
	Size	Mode	File
CRM Buffer	191	Lisp Interaction	
[], *scratch*	18084	Fundamental	/shared/SparkBuild.scala
% *GNU Emacs*	698	Fundamental	
%* *Messages*	157	Messages	
%* *Warnings*	300	Special	



Concepts: Mapeamento

- Classificação de mensagens

Gmail



overloaded concept

Concepts: Mapeamento

- Configurar “Paper Size” em drivers de impressoras

