

# Interface & Interaction Design

Vorlesung: Grundlagen & Designkonzepte

VU 183.289

WS 2016/2017

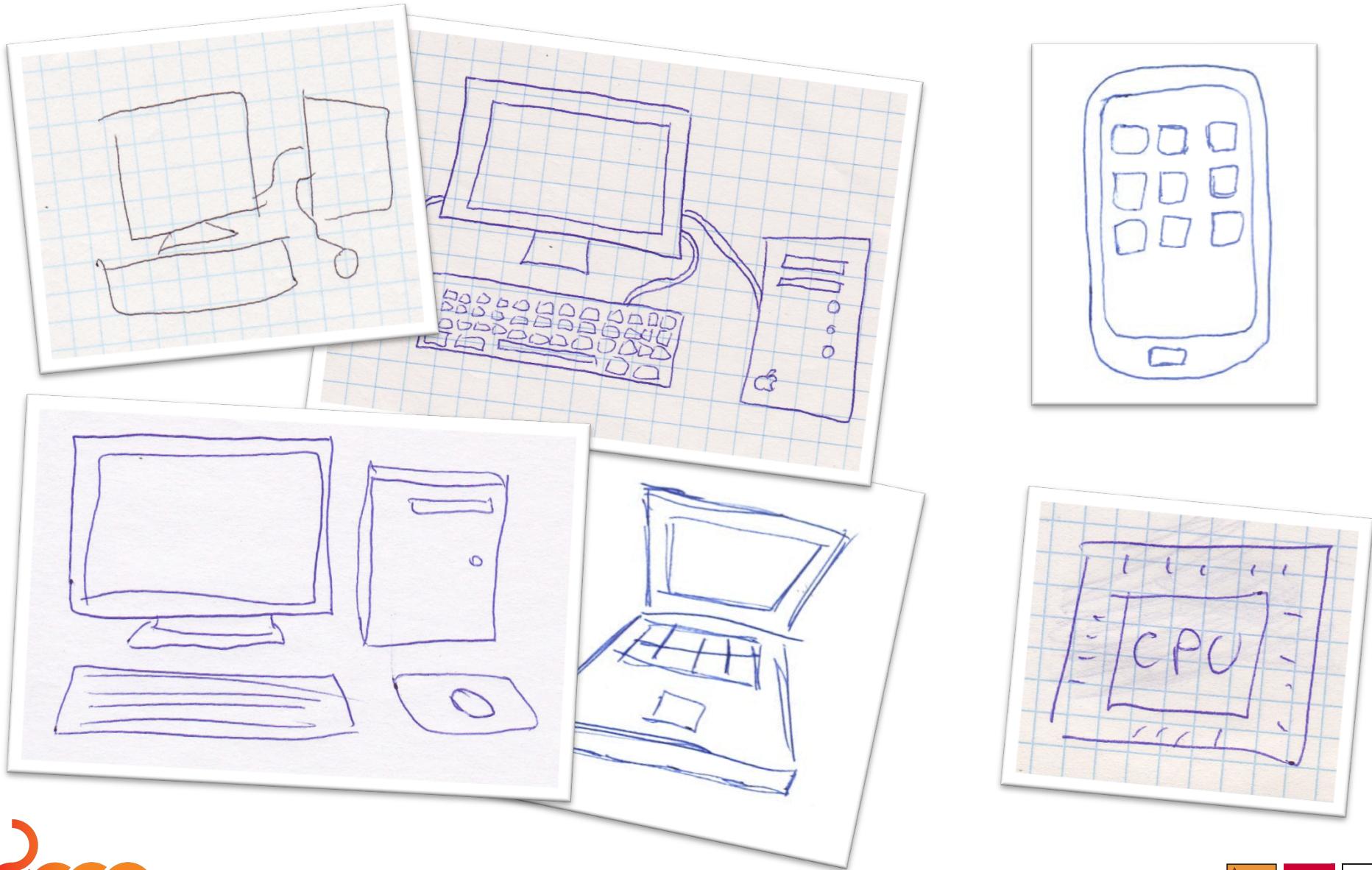
Christoph Wimmer



# Überblick

- Ein kurzes Experiment
- Interaction Framework
- Grundlegende Designkonzepte nach Norman
  - Konzeptuelles Modell
  - Affordances & Signifier
  - Mapping
  - Conventions & Consistency
  - Constraints
  - Feedback

# Zeichnen Sie einen Computer in 60 Sekunden



# Tool vs. Interface

Tool



Interface to tool



# Tool vs. Interface

Die Gesamtheit aller Mittel mit denen Menschen (die User, BenutzerInnen) mit einer Maschine, einem Gerät, einem Computerprogramm oder einem anderen komplexen Tool (dem System) interagieren.

# Tool vs. Interface

- Das User Interface ist nicht nur das, was man auf dem Bildschirm sieht
- Man hat als UI DesignerIn in der Regel keine vollständige Kontrolle über das User Interface
- Man muss sich als UI DesignerIn über die unterschiedlichen Rahmenbedingungen bewusst sein und diese berücksichtigen





Personal Computer



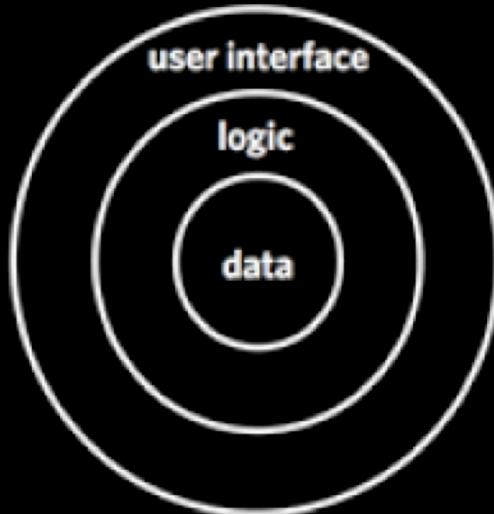
Spielkonsole



Smartphone

Computer, noun: an electronic device which is capable of receiving information (data) in a particular form and of performing a sequence of operations in accordance with a predetermined but variable set of procedural instructions (program) to produce a result in the form of information or signals. – Oxford Dictionary

Your program



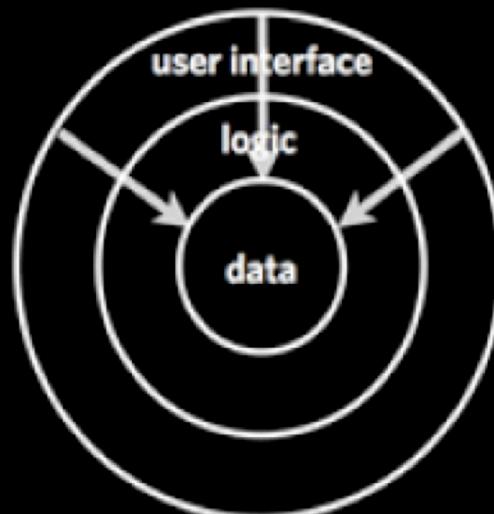
How users see it



How you see it



How you should see it



Quelle: [http://www.nickfinck.com/blog/entry/the\\_ten\\_commandments\\_of\\_user\\_experience/](http://www.nickfinck.com/blog/entry/the_ten_commandments_of_user_experience/)

Die Benutzerschnittstelle und Art der Interaktion beeinflusst stark unsere Vorstellung von Computern.

Durch eine Änderung der Benutzerschnittstelle können DesignerInnen und EntwicklerInnen maßgeblichen Einfluss auf das mentale Modell über die Funktionsweise eines Systems von BenutzerInnen ausüben.

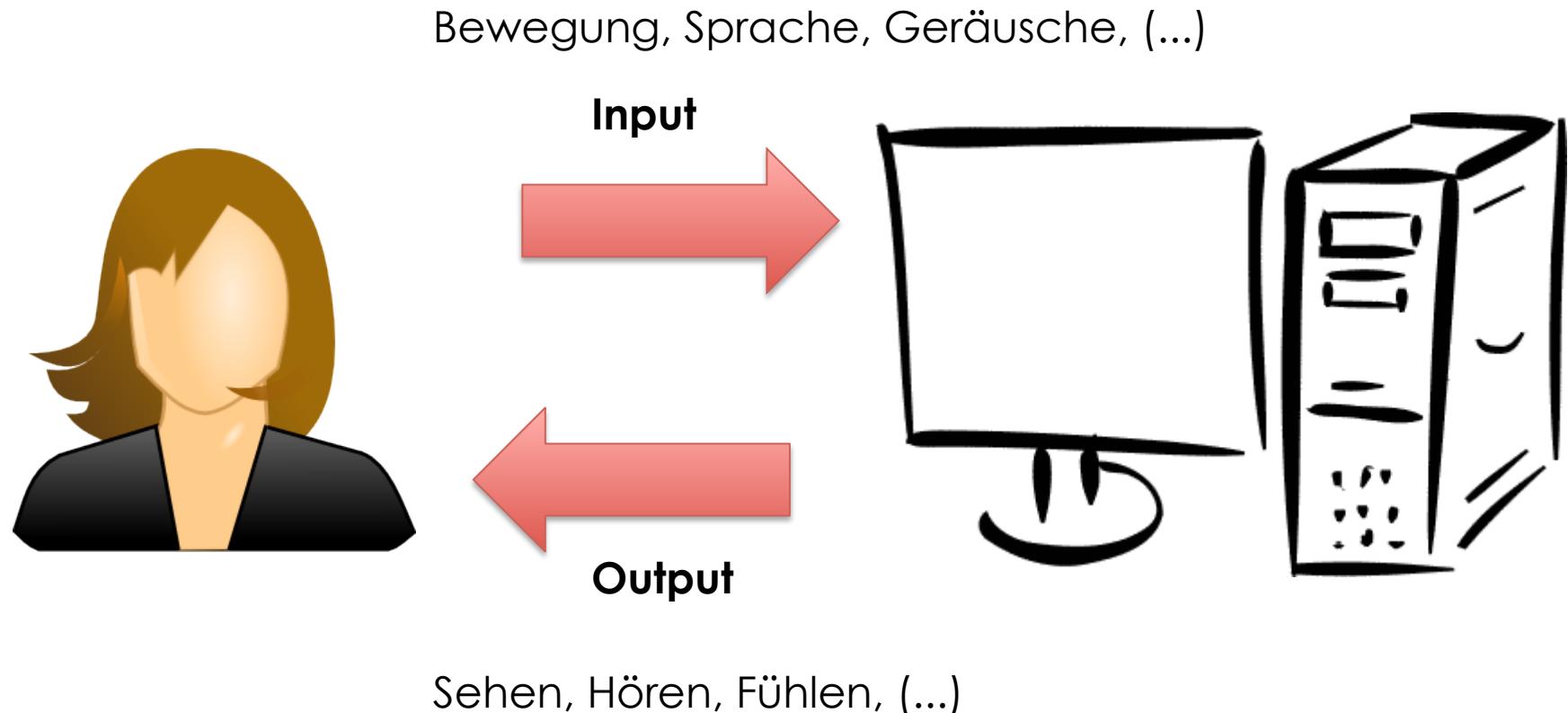
# Interaction Framework

# Interaction: Definition

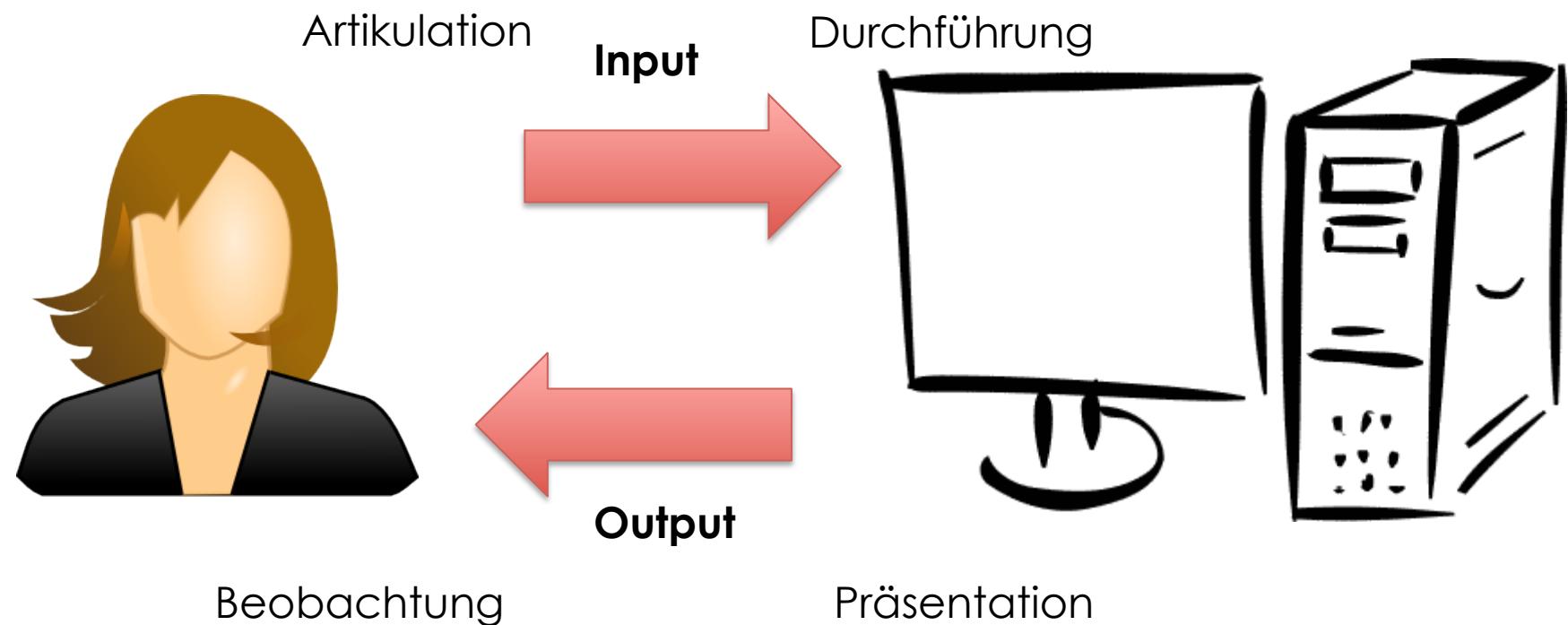
Interaction, noun: Reciprocal action or influence

Oxford Dictionary

# Interaction Framework



# Interaction Framework



Adaptiert nach Dix et al.

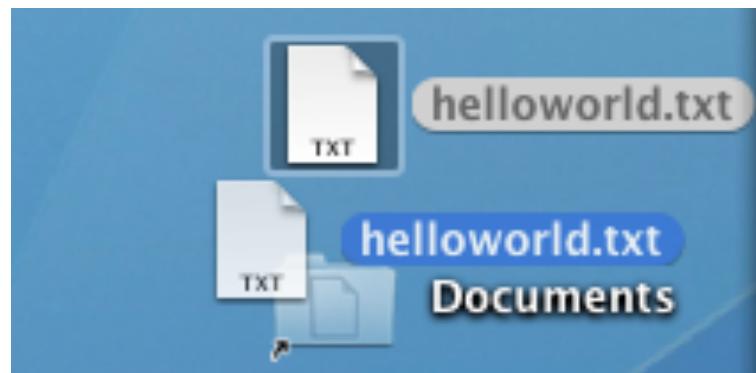
# Interaction Framework

- Ausführungs-Phase
  - Artikulation: Der/die BenutzerIn übersetzt seine/ ihre Absicht in die Eingabesprache des Systems
  - Durchführung: Das System übersetzt die Eingabe in Operationen, die vom System ausgeführt werden können
  - Präsentation: Das System stellt das Ergebnis der Operationen mittels einer Ausgabesprache dar
- Evaluierungs-Phase
  - Beobachtung: Der/die BenutzerIn interpretiert die präsentierten Ergebnisse und vergleicht sie mit seiner/ihrer ursprünglichen Zielsetzung

# Beispiel: Datei verschieben

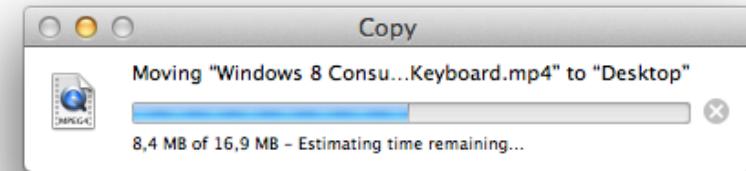
Artikulation

```
mv helloworld.txt ./Documents
```

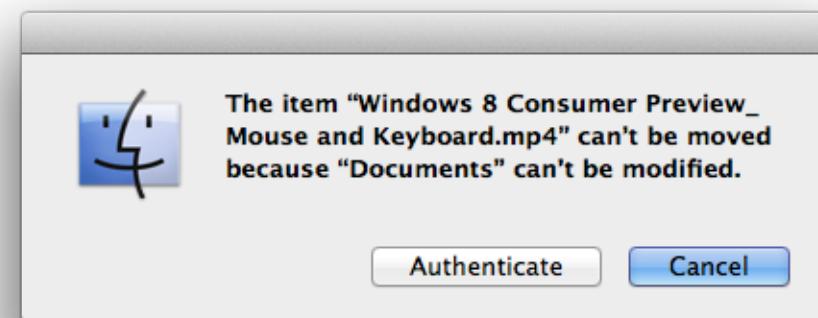


„Verschiebe die Datei ...“

Präsentation



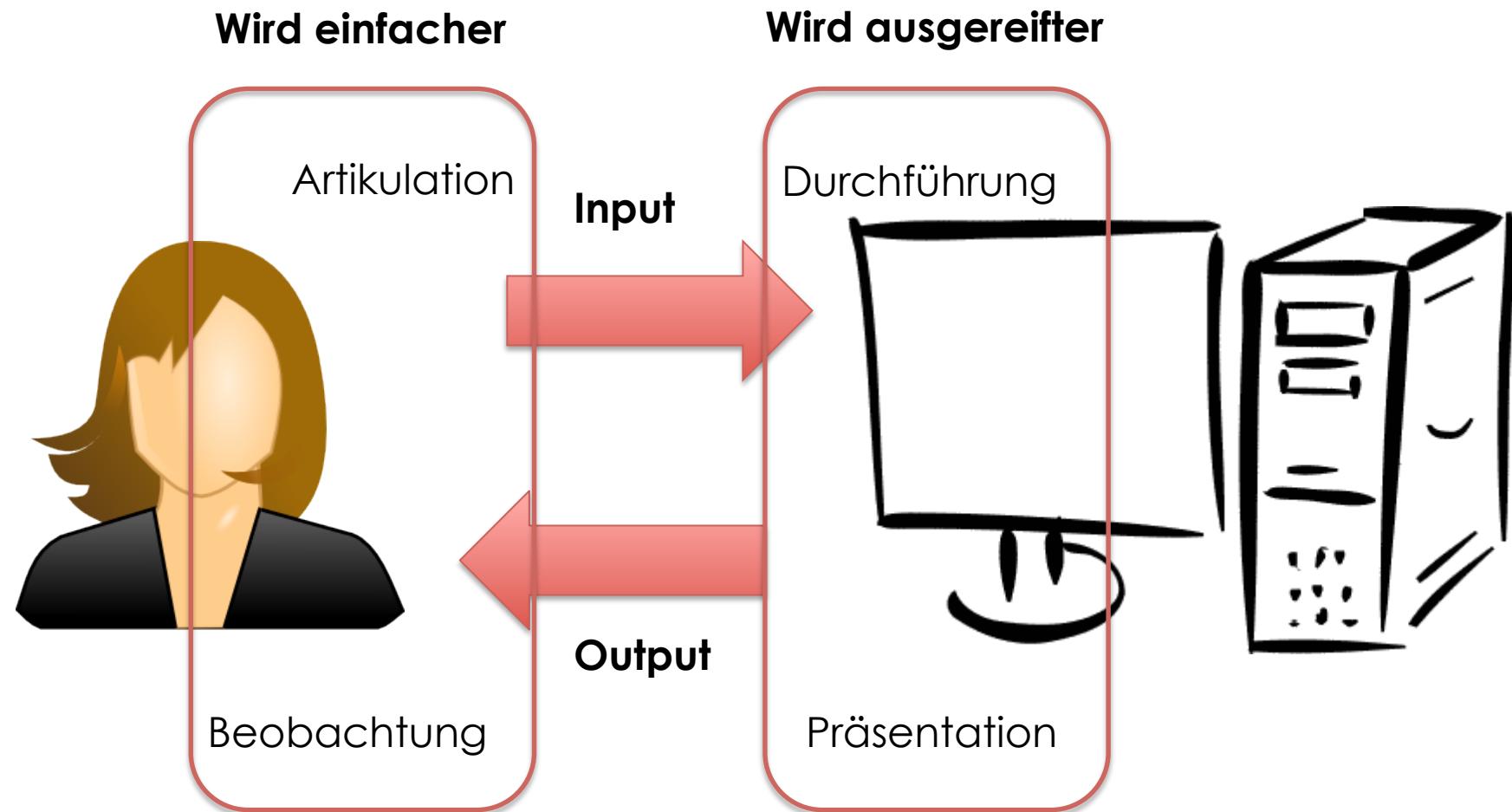
\*Pling\*



Gutes Interface & Interaction Design macht dem/  
der BenutzerIn die Artikulation seiner/ihrer Absichten  
möglichst einfach und schnell.

Gutes Interface & Interaction Design macht dem/  
der BenutzerIn die Beobachtung und Interpretation  
des Systemzustands klar und verständlich.

# Interaction Framework

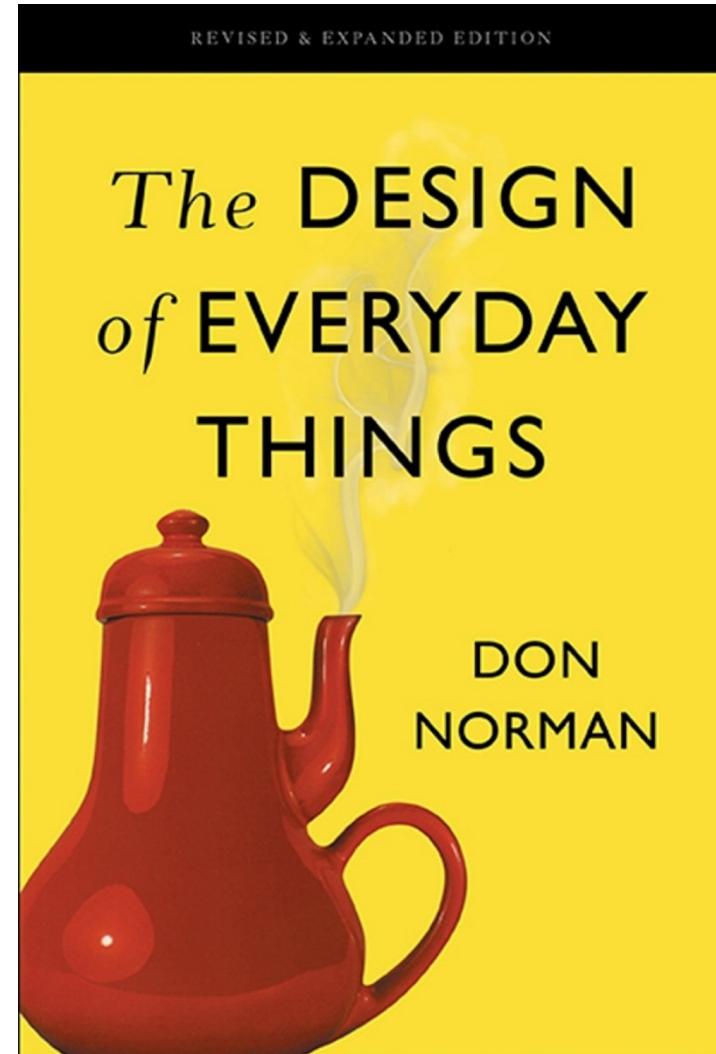


Adaptiert nach Dix et al.

# Designkonzepte nach Donald Norman

# Design Konzepte

1. Konzeptuelles Modell
2. Affordances & Signifier
3. Mapping
4. Conventions & Consistency
5. Constraints
6. Feedback



# Konzeptuelles Modell

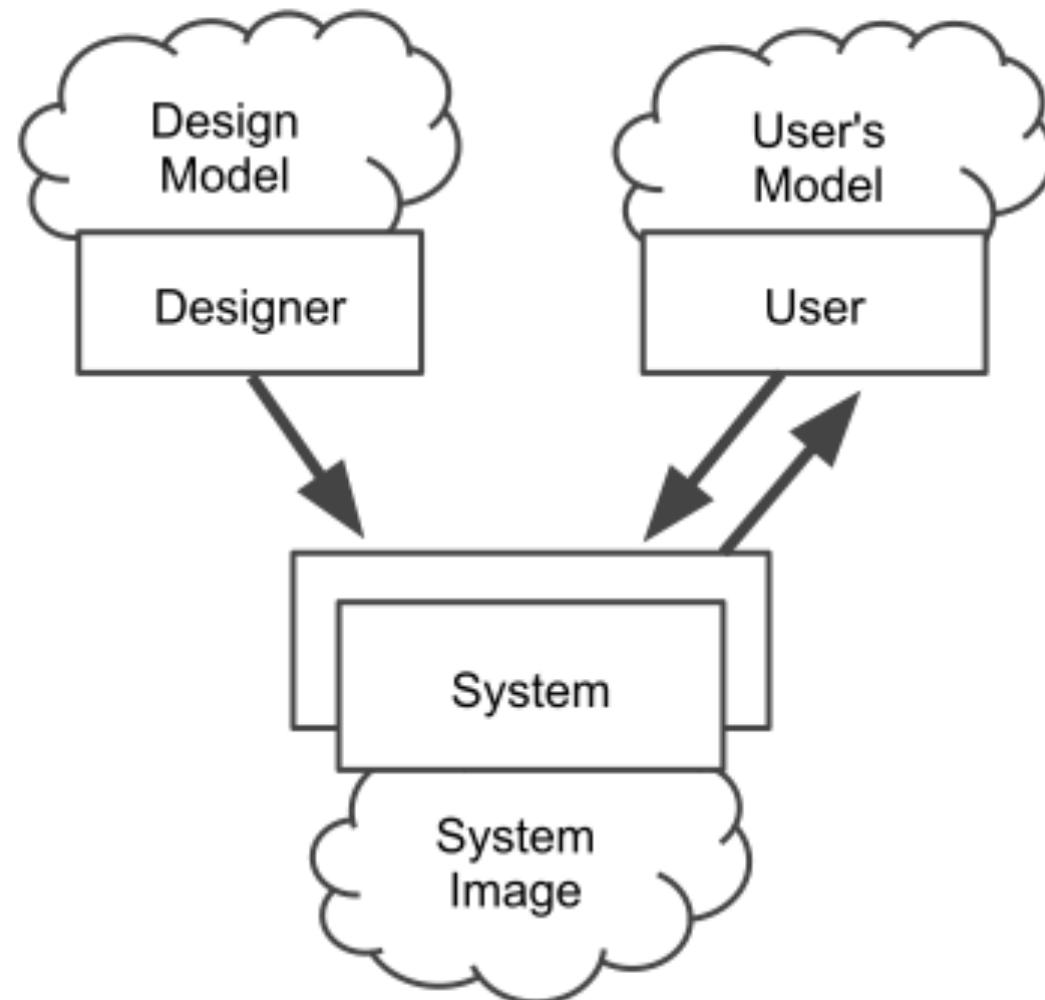
- Die BenutzerInnen haben ein mentales Modell von der Funktionsweise von Dingen
- Modelle erlauben uns die Operation von einem Gerät mental durchzuspielen
- Dieses Modell entsteht durch
  - Affordances & Signifier
  - Mapping
  - Conventions & Consistency
  - Constraints
  - Feedback
- Schlechtes Design kann zu falschen mentalen Modellen führen

# Konzeptuelles Modell



Jaque Carelman, Impossible Objects

# Konzeptuelles Modell



# Konzeptuelles Modell

„Indeed, the models that learners spontaneously form are incomplete, inconsistent, unstable in time, and often ripe with superstition.“ (Olson & Carroll)

„All models are wrong, but some are useful“ (G. Box)

„If your program model is non trivial, it's probably not the user model.“

„A user interface is well-designed when the program behaves exactly how the user thought it would.“ (Joel Spolsky)

# Konzeptuelles Modell

- Was passiert, wenn wir auf Google suchen?
  
- Ich gebe einen Suchbegriff in das Eingabefeld ein
- Ich drücke Enter
- Ich bekomme eine Liste mit Suchergebnissen angezeigt
  
- Vergleiche:  
<https://github.com/alex/what-happens-when>

# Knowledge in the Head

- Erinnern/Merken von Funktionsabläufen, Dingen oder Ereignissen ist Knowledge in the Head
- Muss erlernt werden, ist aber sehr effizient
- Beispiele:
  - Namen
  - Passwörter
  - Sozialversicherungsnummer
  - Geburtstage
  - Adressen?
  - Telefonnummern?
  - E-Mail Adressen?
  - ...

# Knowledge in the World

- Wissen, das aus der Welt abgeleitet werden kann
- Muss nicht erlernt werden, ist aber weniger effizient,  
da die externe Information gesucht und interpretiert  
werden muss
- Beispiele:
  - Terminkalender (persönlich, privat)
  - Knoten im Taschentuch (Signal vs. Nachricht)
  - Labels, Zeichen, Texte (universell, öffentlich)
  - Natural Mappings (Beispiel: Herdplatten)



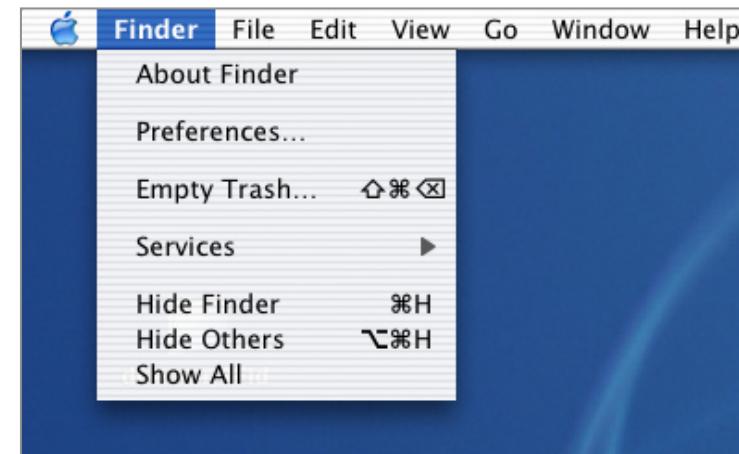
# Recall vs. Recognition

```
A:\>dir Cs105data

Volume in drive A has no label
Volume Serial Number is 032B-0092
Directory of A:\Cs105data

.           <DIR>      03-09-98 11:42p .
..          <DIR>      03-09-98 11:42p ..
BOTTOM     DOC      14,848  06-23-97  2:14p Bottom.DOC
TOP        DOC      14,336  06-23-97  2:13p Top.DOC
HANDED    TXT       5,155  07-16-97  3:58p HANDED.TXT
WORLD     XLS     18,432  07-12-97  1:55a world.xls
               4 file(s)      52,771 bytes
               2 dir(s)      674,816 bytes free

A:\>_
```



# Two Kinds of Errors

- Slips
  - Treten trotz korrektem konzeptuellen Modell auf
  - Uns ist bewusst, was falsch gelaufen ist und wie wir den Fehler beheben können
  - Ursache in der Ausführung, z.B. weil wir überhastet arbeiten, Kontrollelemente zu nahe beisammen liegen, Unterbrechungen, ...
- Mistakes
  - Treten wegen falschem konzeptuellen Modell auf
  - Wir wissen nicht, was falsch gelaufen ist und wie wir den Fehler beheben können
  - Ursache in der Planung, weil wir eine falsche Vorstellung von der Funktionsweise haben

# Affordances & Signifier

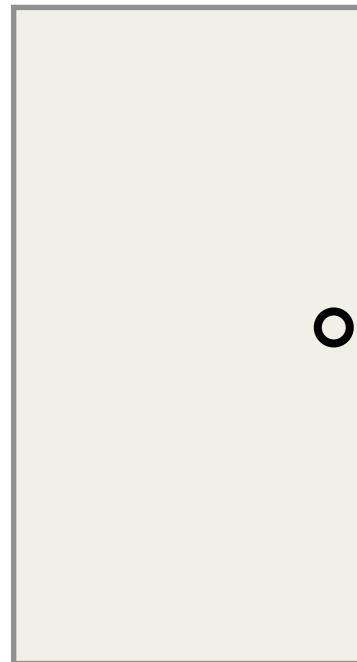
- „[A]ffordance refers to the **perceived** and **actual properties** of the thing, primarily those fundamental properties that determine just **how the thing could possibly be used.**“
- Aufforderungscharakter: Das Erscheinungsbild sollte die Verwendung des Objektes und Handlungsmöglichkeiten des Benutzers / der Benutzerin suggerieren.



# Affordances von Türen

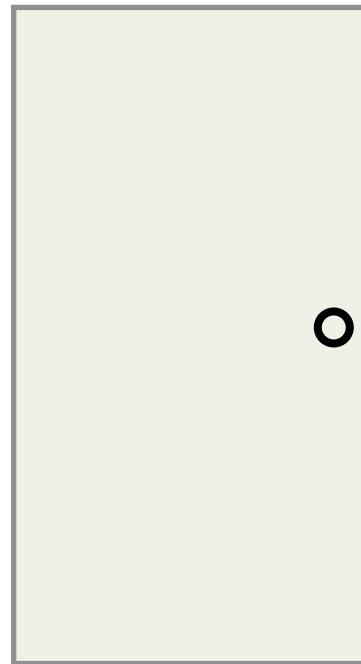


# Affordances von Türen

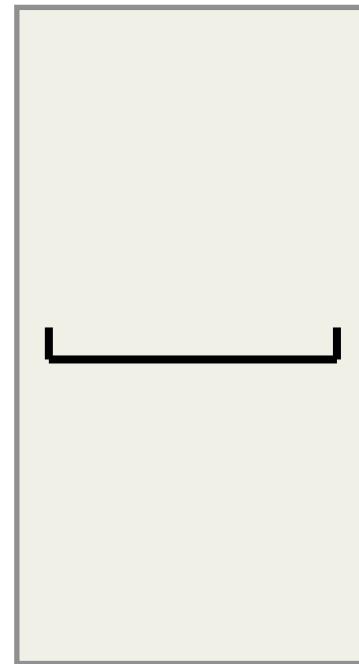


Drücken oder Ziehen?

# Affordances von Türen

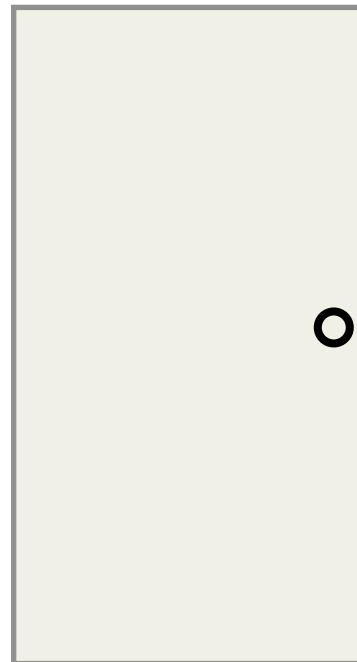


Drücken oder Ziehen?

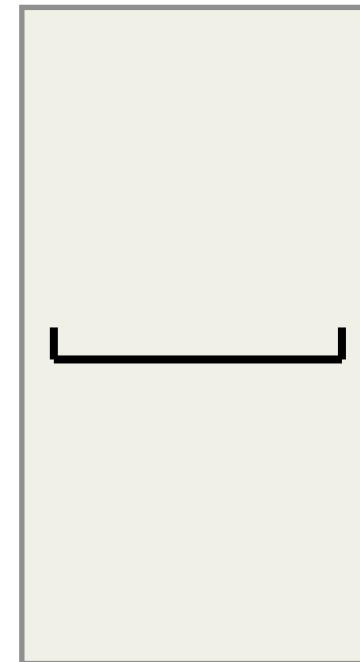


Welche Seite?

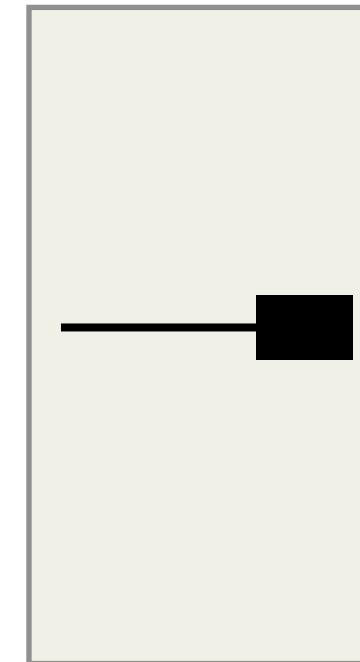
# Affordances von Türen



Drücken oder Ziehen?



Welche Seite?



Kann nur gedrückt werden  
Klar ersichtlich auf welcher  
Seite

# Affordances

- Komplexe Dinge benötigen oft eine Anleitung
- Einfache Dinge sollten keine benötigen
- Brauchen einfache Dinge zur Erklärung ihrer Funktion Labels oder Instruktionen, so ist das Design meist gescheitert



# Affordances vs. Signifier?

The concept of "**affordance**" [...] was originally invented by the perceptual psychologist J. J. Gibson to refer to a relationship: the actions possible by a specific agent on a specific environment. To Gibson, affordances **did not have to be perceivable or even knowable** -- they simply existed. When I introduced the term into design in 1988 I was referring to **perceivable affordances**.

(Donald Norman)

# Affordances vs. Signifier?

Affordances exist even if they are not visible. For designers, their visibility is critical[.]

If an affordance [...] cannot be perceived, some means of signaling its presence is required: I call this property a **signifier**.

Affordances determine what **actions are possible**. Signifiers communicate **where the action should take place**. We need both.

(Donald Norman)

# Affordances vs. Signifier?

- **Affordance:** Handlungsmöglichkeiten, die ein Objekt einem/einer Akteurln zur Verfügung stellt (unabhängig ob wahrnehmbar oder nicht)
- **Signifier:** Die Repräsentation der vorhandenen Handlungsmöglichkeiten, wodurch diese für den/ die Akteurln wahrnehmbar werden
- **Perceived Affordance:** Von einem/einer Akteurln wahrgenommene Handlungsmöglichkeiten
- Bisher wurde nur selten ausdrücklich zwischen diesen Konzepten unterschieden – meist ist damit allgemein die **Wahrnehmbarkeit von Handlungsmöglichkeiten** gemeint

# Affordances vs. Signifier?

An affordance is a **relationship** between the **properties of an object** and the **capabilities of the agent** that determine just how the object could possibly be used.

The presence of an affordance is jointly determined by the **qualities of the object** and the **abilities of the agent** that is interacting.

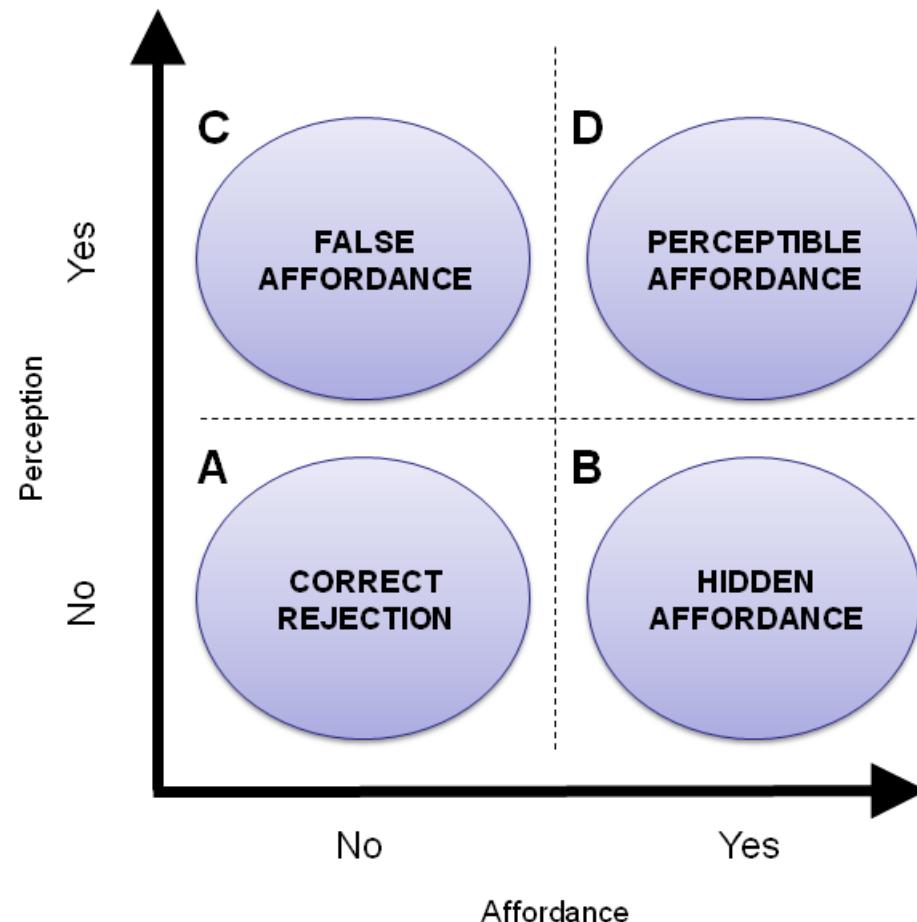
We are used to thinking that properties are associated with objects. But affordance is not a property. An **affordance is a relationship**. Whether an affordance exists depends upon the properties of both the object and the agent.

(Donald Norman)

# Affordances & Signifier

- Warum sind Signifier für Software User Interface Design wichtig?
- Software Interfaces sind virtuell und weisen daher keine inhärenten, physischen Affordances auf
- Software User Interfaces können eine fast beliebige Gestalt annehmen, sie sind an keine Form gebunden
- Die wahrgenommene Affordance ergibt sich nur aus explizit gestalteten Signalen

# Affordances & Signifier



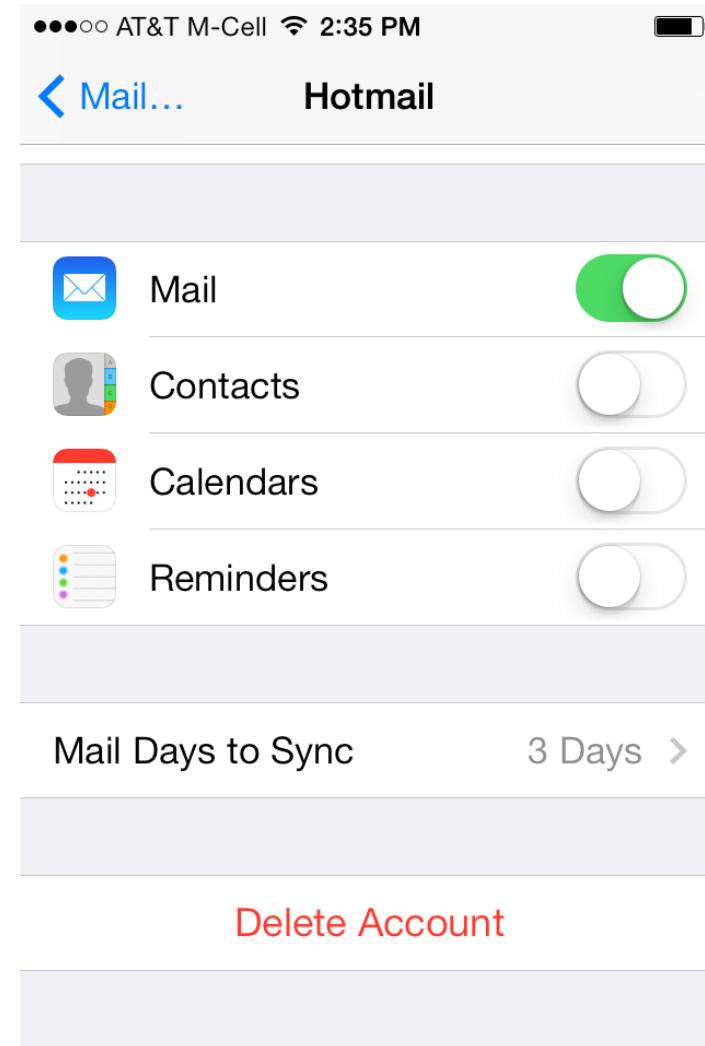
Nach W. Gaver (1991)

# Affordances & Signifier

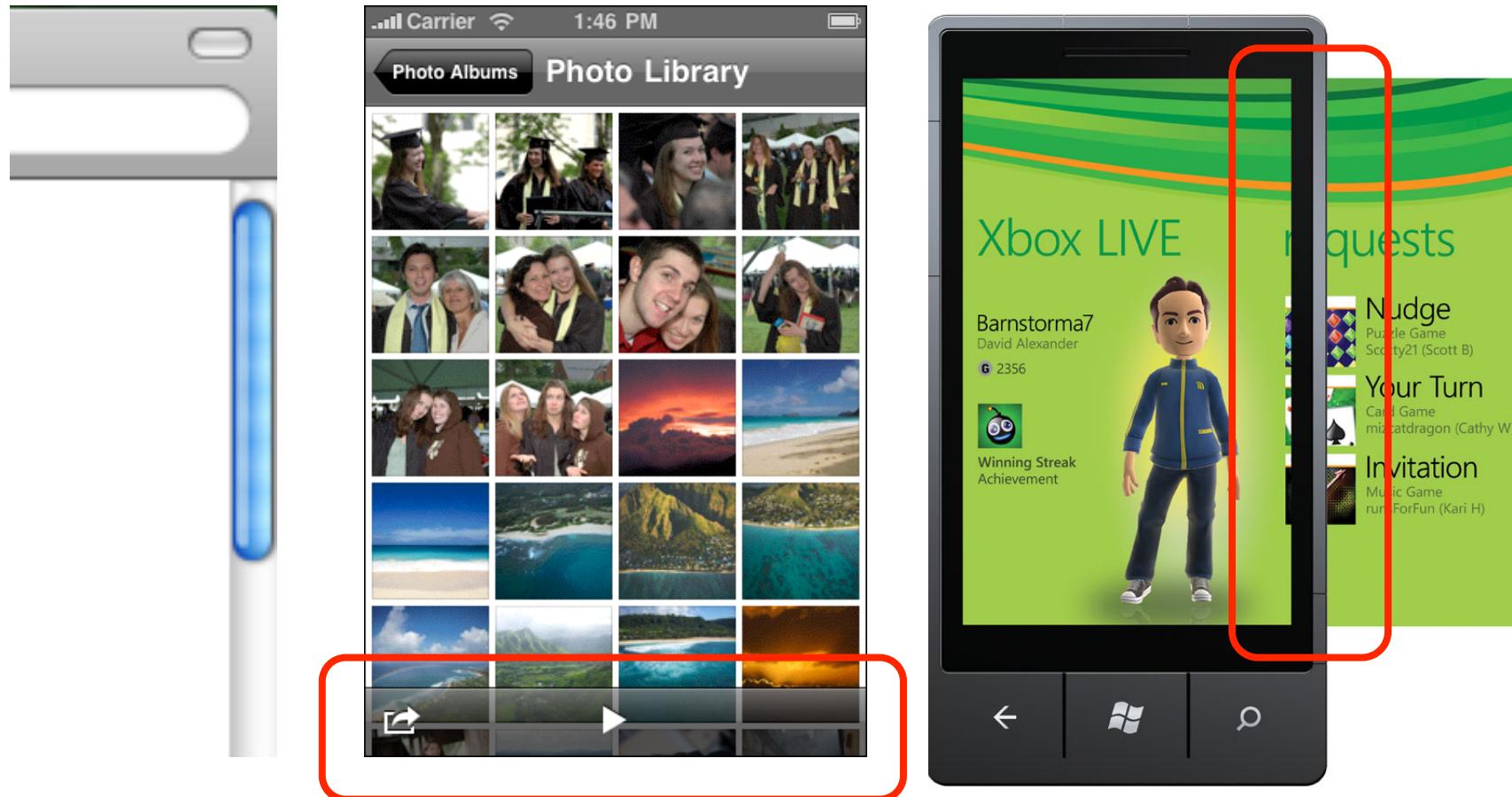


Quelle: Neven Mrgan

# Affordances & Signifier



# Affordances & Signifier





[deco.inso.tuwien.ac.at](http://deco.inso.tuwien.ac.at)

[deco@inso.tuwien.ac.at](mailto:deco@inso.tuwien.ac.at)