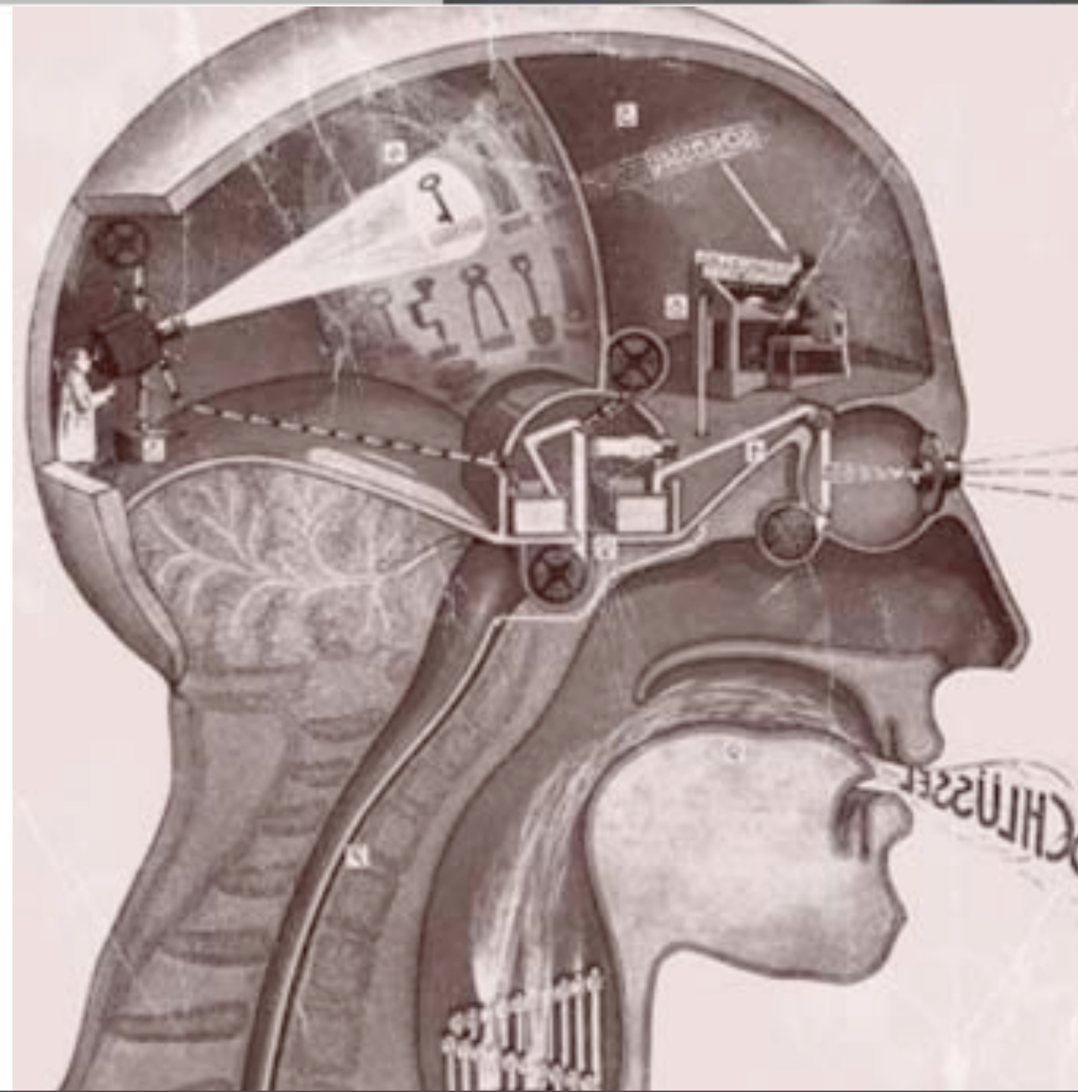
Human capabilities: Memory and learning

HCI Lecture 3

Mark Wright



Key points:

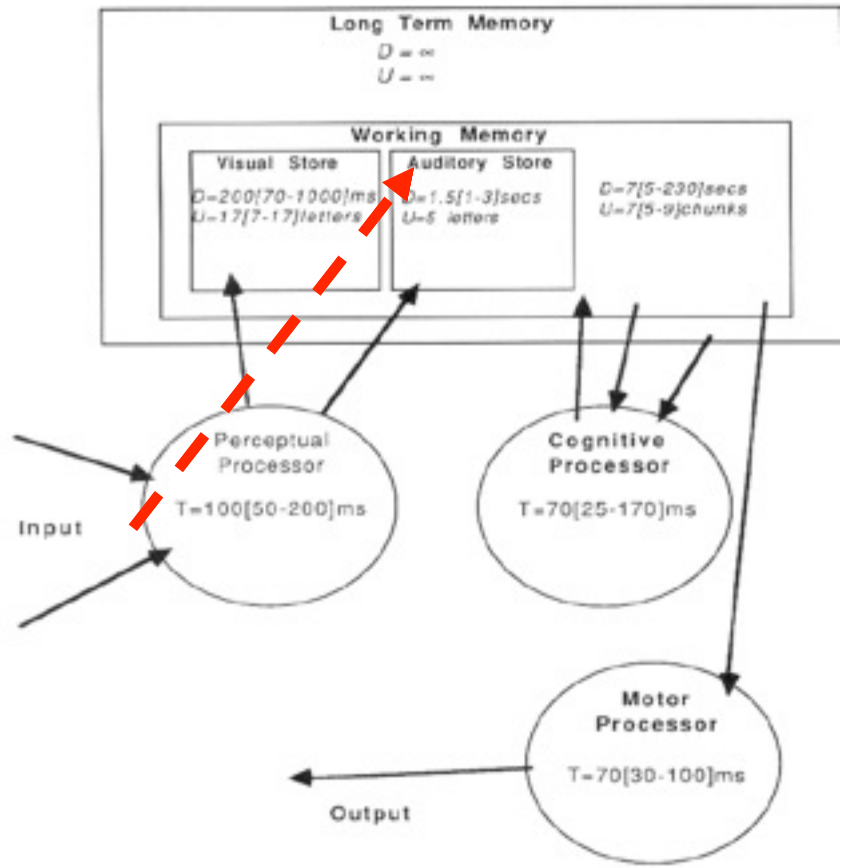
- Long term memory
  - organised as networks, schemas, frames
- Short term memory
  - what are the limits?
- Learning rates
- Learnability

Human constraints

The MHP

- **Model Human Processor** (MHP) (Card, et al., 1983)
- Perceptual, Motor and Cognitive sub-systems characterised by:
  - Storage capacity *U*
  - Decay time *D*
  - Processor cycle time *T*

- We will focus today on the memory stores

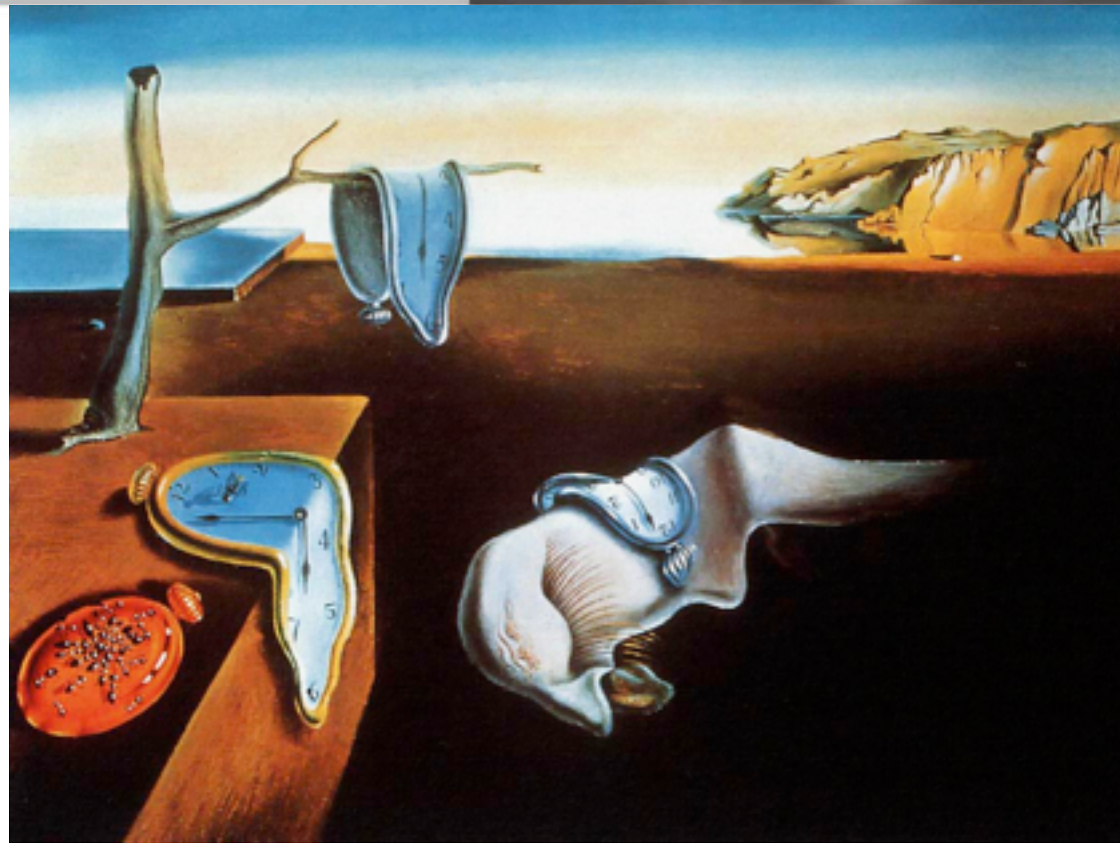


Long term memory

Long Term Memory (LTM)

Infinite capacity and decay time?

- Not everything is stored
  (what is filtering process?)
- Not everything stored can be retrieved
  (what is recall process?)
- Not everything recalled is correct
  (what is interference process?)



The Persistence of Memory
Salvador Dali

Long term memory

Different kinds of Memory:

Declarative, knowledge of facts:

Episodic:

what happened,

where and when

Long term memory

Different kinds of Memory:

Declarative, knowledge of facts:

Semantic:

factual information,

general knowledge

independent of context

Long term memory

We can define different kinds of Memory:

Procedural: *how-to-do-it* knowledge

Usually implicit, hard to put in words

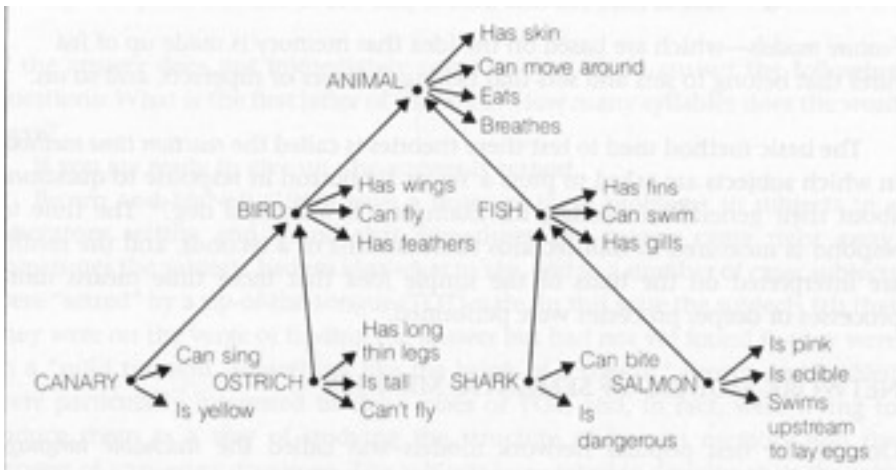
hence ‘non-declarative

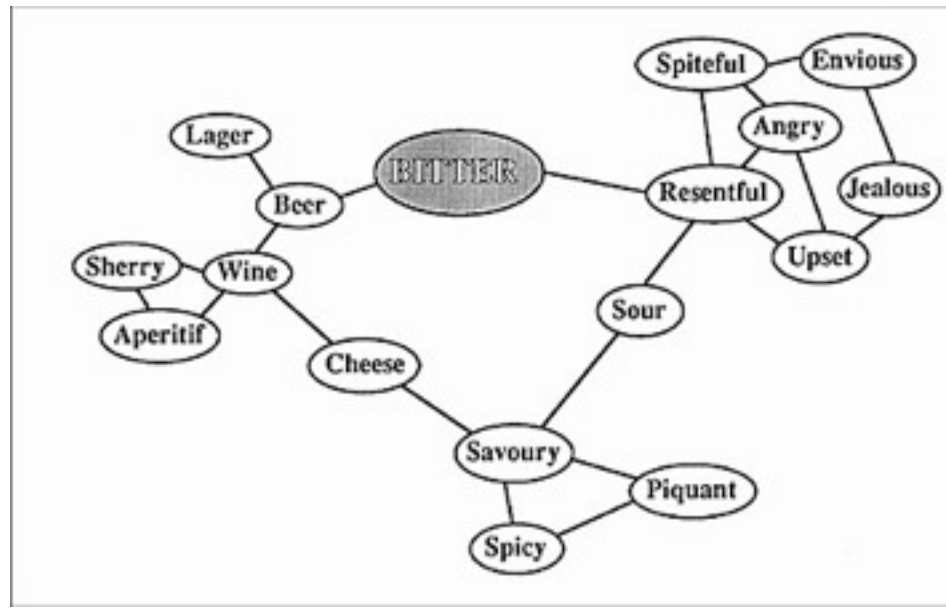
e.g. how to ride a bicycle

Declarative memory models

Semantic nets: memory is organised by links expressing strength or type of relationships between nodes

May be hierarchical

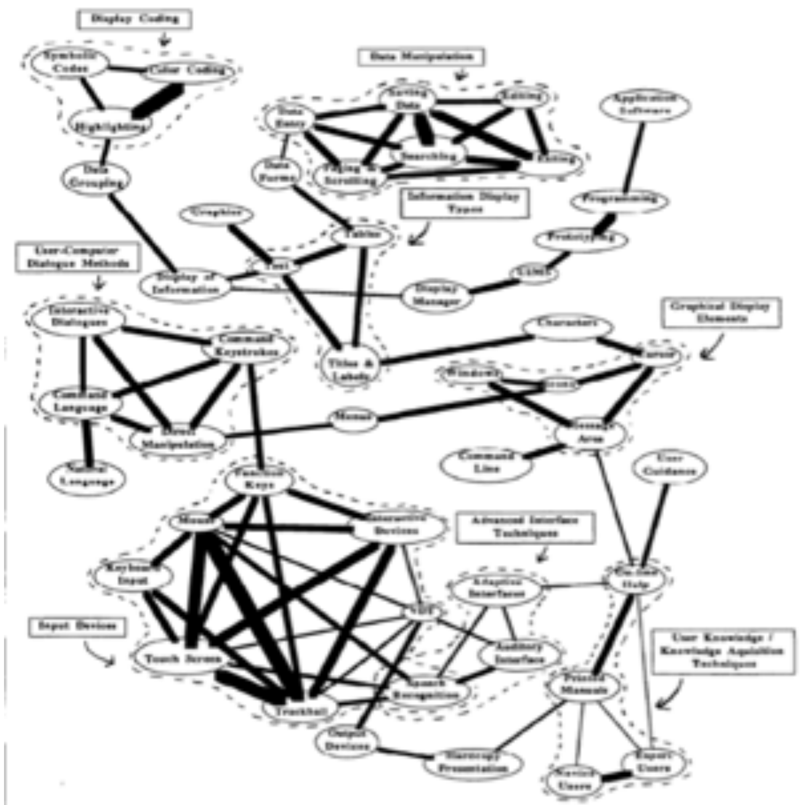




Can generate representation of people's knowledge by asking them to rank relatedness of item pairs, then generate and prune network (e.g. Pathfinder algorithm)

Declarative memory models

Human factors specialists



Software developers

(n.b. novices had relatively unstructured network)

Designers' models of the Human-Computer Interface Gillan & Breedin (CHI 1990)

Declarative memory models

Schemas: pre-exisiting knowledge structures that shape our memory of new inputs

- Can improve recall but also cause memory biases

The schema concept has been formalised as:

- **Frames:** knowledge is organised into data structures with fixed, default and variable slots or attributes (c.f. OOP)

- **Scripts:** stereotypical knowledge about situations that allows interpretation of partial descriptions or cues
    - E.g. “We went to that restaurant you recommended. The food arrived quickly. We left about nine.” *Did they eat the food? Did they pay? Were there tables in the restaurant?*

Note interaction with episodic memory

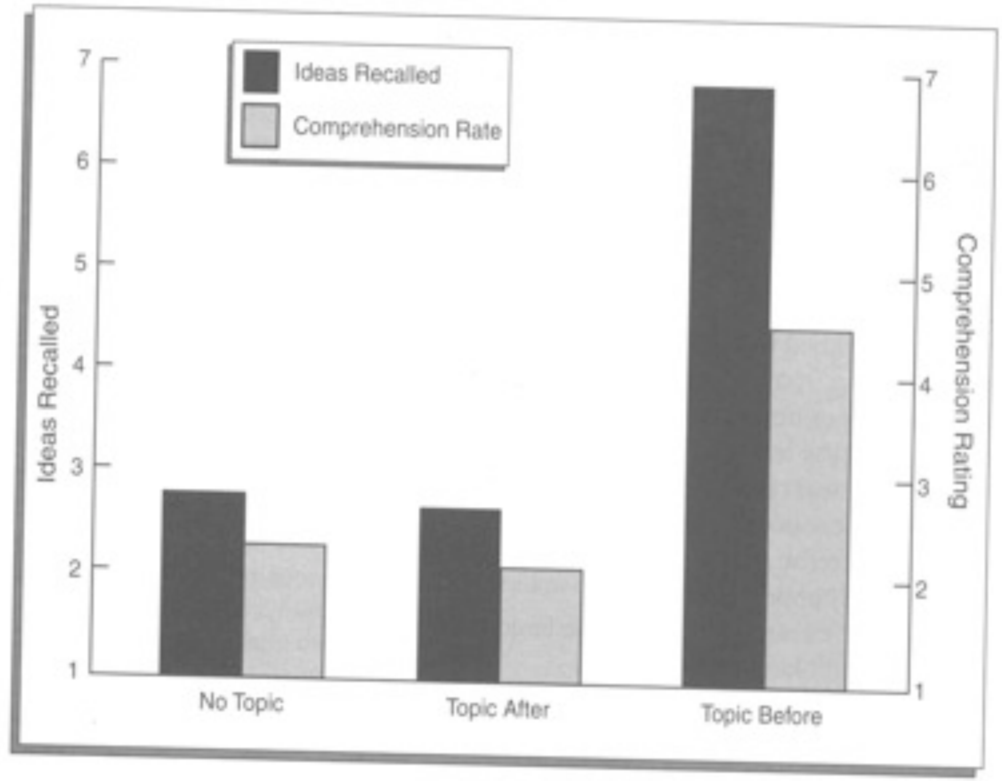
- Schemas may develop as abstractions of specific experiences
- Our memory of specific experiences may be shaped by schema

The use of a schema to create Context

Can you understand and remember what is going on here?

The procedure is actually quite simple. First you arrange items into different groups. Of course one pile may be sufficient depending on how much there is to do. If you have to go somewhere else due to lack of facilities, that is the next step; otherwise, you are pretty well set. It is better not to overdo things. That is, it is better to do too few things at once than too many. In the short run this may not seem important, but complications can easily arise. A mistake can be expensive as well. It is difficult to foresee any end to the necessity for this task in the immediate future, but then one never can tell.

Context (Bransford & Johnson, 1972)



Recall and comprehension of "washing clothes" story

If you have the correct Schema to hand memory is greatly improved

Procedural memory

'How to do it'

Often modelled in HCI as **production rules**:

Set of rules in the form: *if* **condition** *then* **action**

- **Conditions:** e.g. user goals (and subgoals) and current STM contents
- **Actions:** what to do (and update to goals)

Condition	Action
GOAL: DELETE	[POSITION CURSOR] [PRESS DELETE]

```
(TARGET-IS-LOCATED-BEGIN-MOVING-MOUSE
IF
((GOAL DO MENU TASK)
 (STEP VISUAL-SEARCH)
 (WM TARGET-TEXT IS ?T)
 (VISUAL ?TARGET-OBJECT LABEL ?T)
 (WM CURSOR IS ?CURSOR-OBJECT)
 (MOTOR MANUAL PROCESSOR FREE))
THEN
((DELDB (STEP VISUAL-SEARCH))
 (ADDDB (STEP MAKE RESPONSE))
 (SEND-TO-MOTOR MANUAL PERFORM PLY MOUSE
       RIGHT ZERO-ORDER-CONTROL ?CURSOR-OBJECT ?TARGET-OBJECT)))
```

Procedural memory

Using production rule descriptions:

- Complexity of interaction task can be estimated from number of production rules needed to describe it.
- Time to learn a task can be estimated by how many production rules transfer to it from previous tasks
- Cognitive load can be estimated by how much working memory the conditions assume

Note This does not imply
these rules are consciously understood

Long term memory

Important constraint is ability to ***retrieve*** information

- Semantic nets imply easier if have cues that are near links to the target
- Schemas imply easier if target is part of coherent structure

Have a much larger capacity for *recognition* than *recall*

- Hence menus vs. command interfaces
- But scanning also takes time, or may have many more items than can be realistically scanned
  - Need to recall where to look for the item to recognise
  - Important to support partial recall (e.g. part of file name)
  - Important to support contextual recall (e.g. when file created)

Short term memory (STM)

Capacity of STM

7±2 chunks of information.

cf. **Working Memory**

- `Registers' of the Cognitive Processor
    - Data from perceptual sub-systems
    - Activated 'chunks' of LTM
- 'Cognitive load' of task is how much we have to keep "in mind"
- Attention bottleneck
- Limited capacity – but what is the limit?

- Common misnomer:

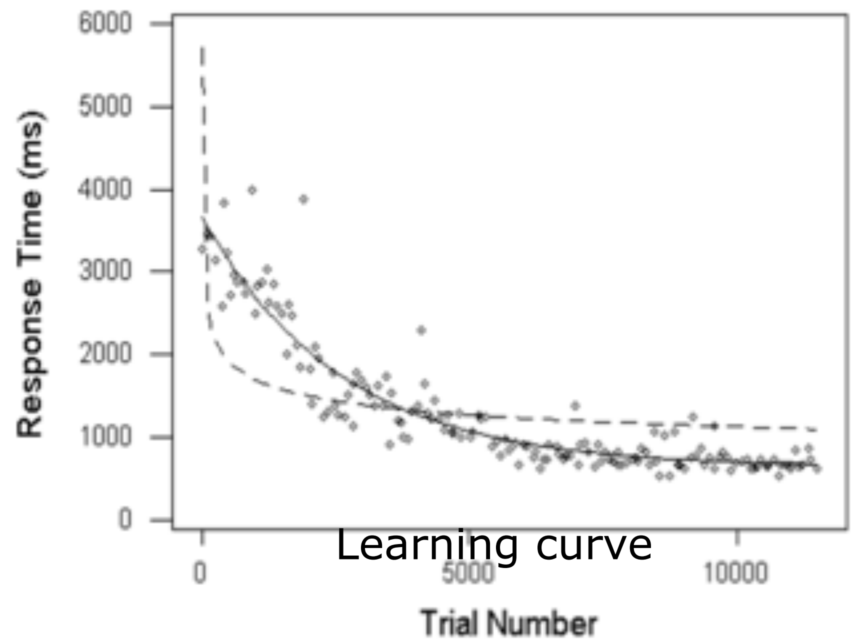
Does not apply directly to menu items - why? - Because we can see them

Learning Rate

Power law of Practice:

- Reaction time: $T_n = T_1 n^{-a}$ $a = 0.4[0.2 -- 0.6]$
- I.e. improvement is rapid at first, and slows later
- Has been found in a wide variety of tasks (pressing button sequences, reading inverted text, mental arithmetic, manufacturing, writing books…)

- However Heathcote et al. (2000) show individual data in a variety of tasks is actually better described as exponential:
  - $T_n = T_1 e^{-an}$
  - Implies constant relative learning rate



Learning curve

Learning - Reinforcement Learning

An alternative framework to the acquisition of ‘production rules’ is the ***reinforcement learning*** approach

Assume an agent is interacting with a world that can be described as a Markov Decision Process

- World contains set of states S
- In each state s the agent can take one of a set of actions A(s)
- Given action a in state s, will have transition to state s’ with probability P(s,s’,a)
- Also have expected reward on the transition R(s,s’,a)

- The problem for the agent is to find a *policy* π for taking the right action in a given state to maximise the expected future reward

Ignoring (for now) the various AI algorithms for solving this problem, we can use it as a framework for understanding what makes interfaces more or less learnable

Learnability (Dix 7.2)

Learnability - The ease with with new users can begin effective interaction and achieve maximal performance

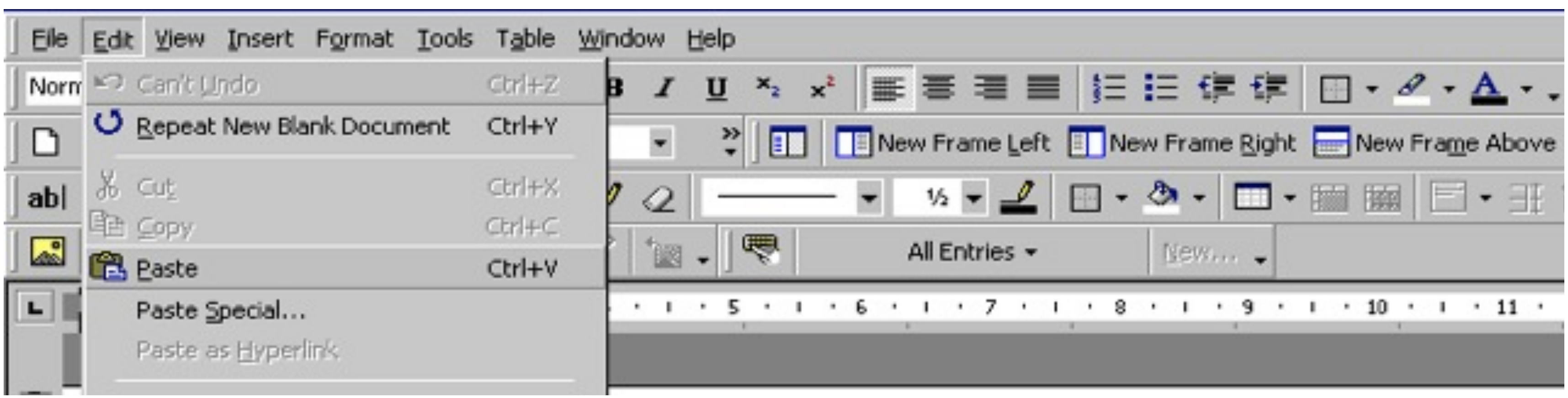
- Predictability
- Synthesizability
- Familiarity
- Generalizability
- Consistency

Learnability (Dix #1)

Predictability — determinism and operation visibility

System behaviour is observably deterministic:

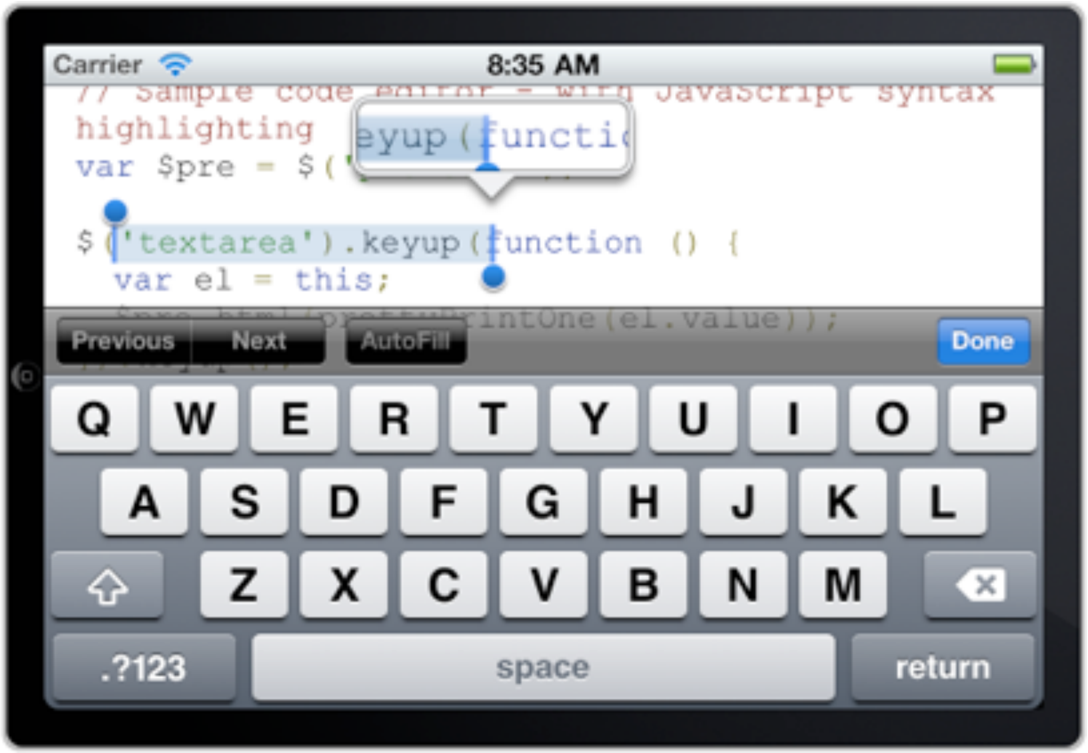
- Easier to learn if *P(s,s',a)=1,* i.e. the same action in the same state has the same consequence
- Also important that user can see that the state has changed as a result of the action (within reasonable delay)
- Markov property: transition does not depend on history (how current state was reached); hence reduced memory load?



Learnability (Dix #1)

Predicability: Support of user to determine the effect of future action based on past interaction history

- Related Principle - Operation Visibility



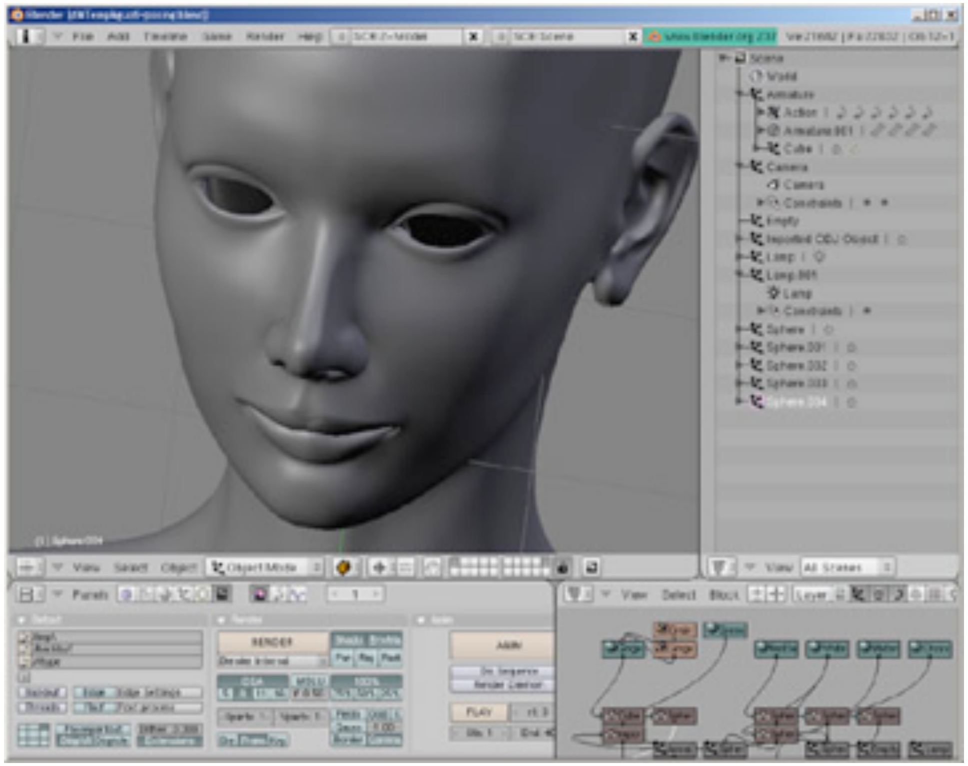
Learnability (Dix #2)

Synthesisability: support for the user to assess the effect of past operations on the current state.

Specifically, they can assess if the outcome is better or worse than expected (are they making progress towards the goal?)

Immediate vs. *eventual honesty*

- Advantage of WYSIWYG - What you see is what you get
- Difficult learning situations involve long chains of states and actions before any reward is received



Learnability (Dix #3)

Familiarity: match the interface to users' expectations:

- Facilitate *guessing*:
  - Suggested that users when guessing will generally pick the action that (superficially) most resembles the goal. Hence should:

1. Make the possible actions salient and distinct, keep number small
2. Use identity cues between actions and goals as much as possible
3. Don't require long sequences of choices
4. Have one or less obscure actions
5. Enable undo.

- Users learn better from exploring, but may be reluctant to explore




Learnability (Dix #3)

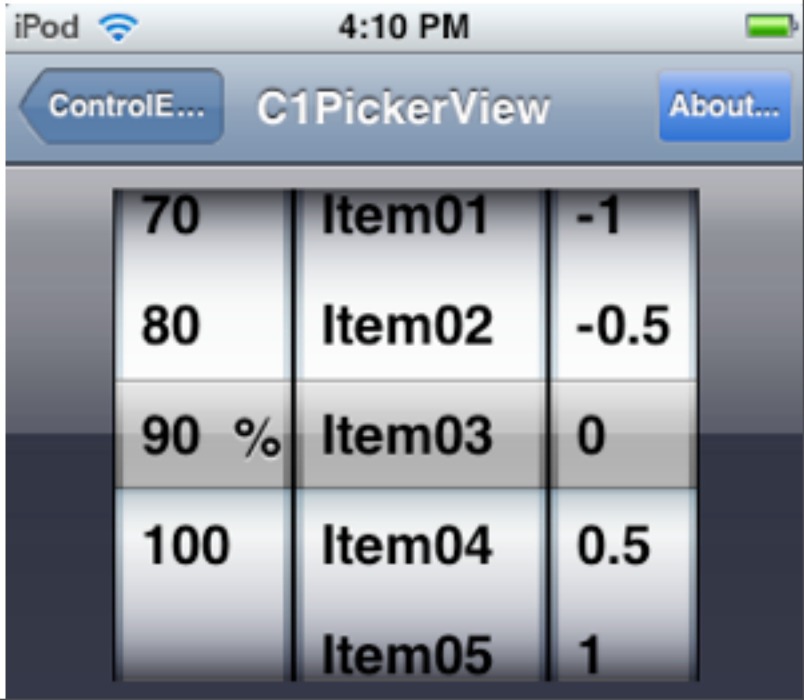
Familiarity: match the interface to users' expectations:

Use terms consistent with everyday usage?

Problem that agreement can be low, e.g. Furnas et al:

- find only 10-20% of users generate same command name as an 'armchair' designer.
- User-preferred names overlap by only 15-35%.
- Up to 15 aliases still covers only 60-80%.

Exploit natural *affordances*



Affordances

Affordance: a relation between agent, object and task

- We don't normally see the world in terms of coloured surfaces in space
- We directly perceive the potential for interaction

"If a terrestrial surface is nearly horizontal (instead of slanted), nearly flat (instead of convex or concave), and sufficiently extended (relative to the size of the animal) and if its substance is rigid (relative to the weight of the animal), then the surface affords support…"

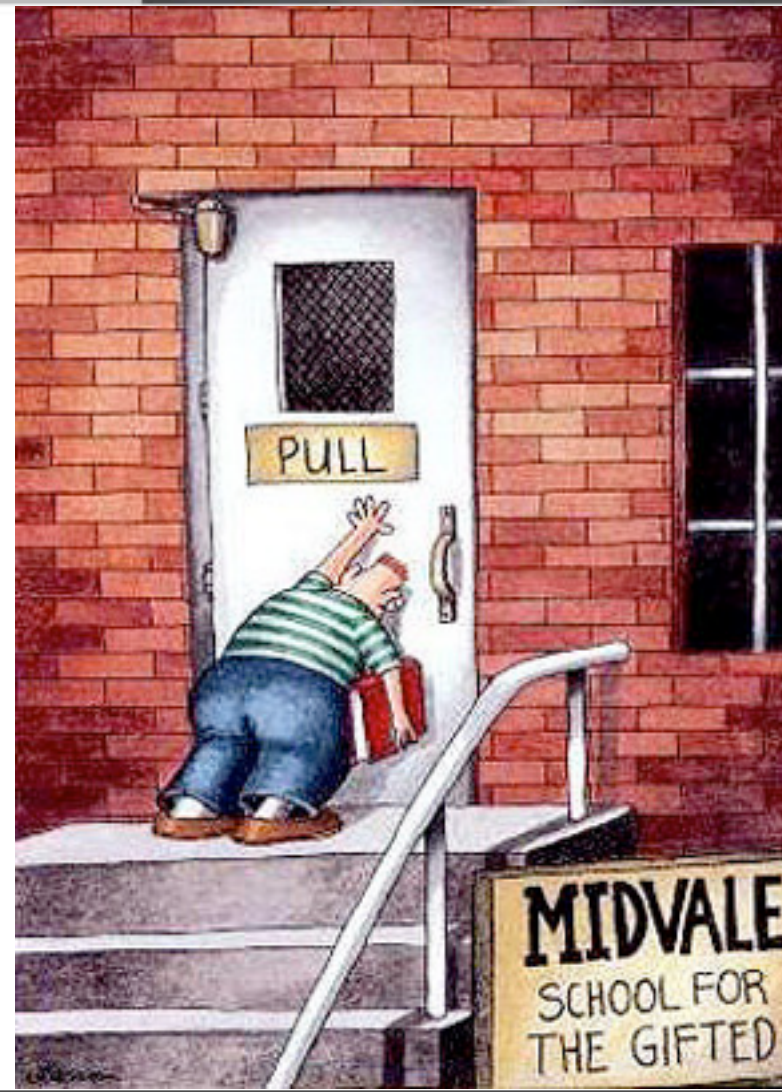
J.J. Gibson (1979) The Ecological Approach to Perception

"If a door handle needs a sign, then its design is faulty"

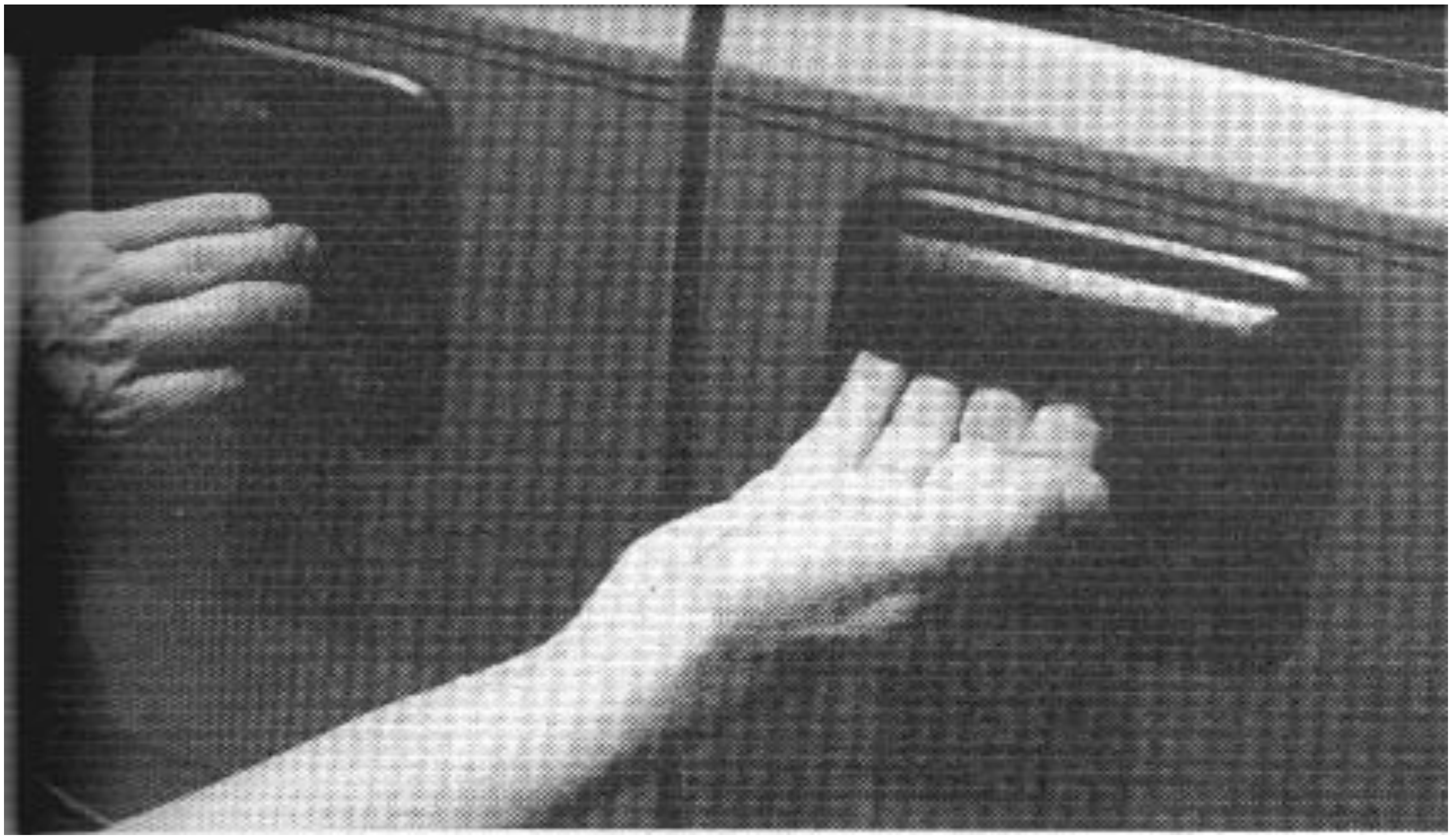
D. Norman, *The Psychology of Everyday Things*

Affordances

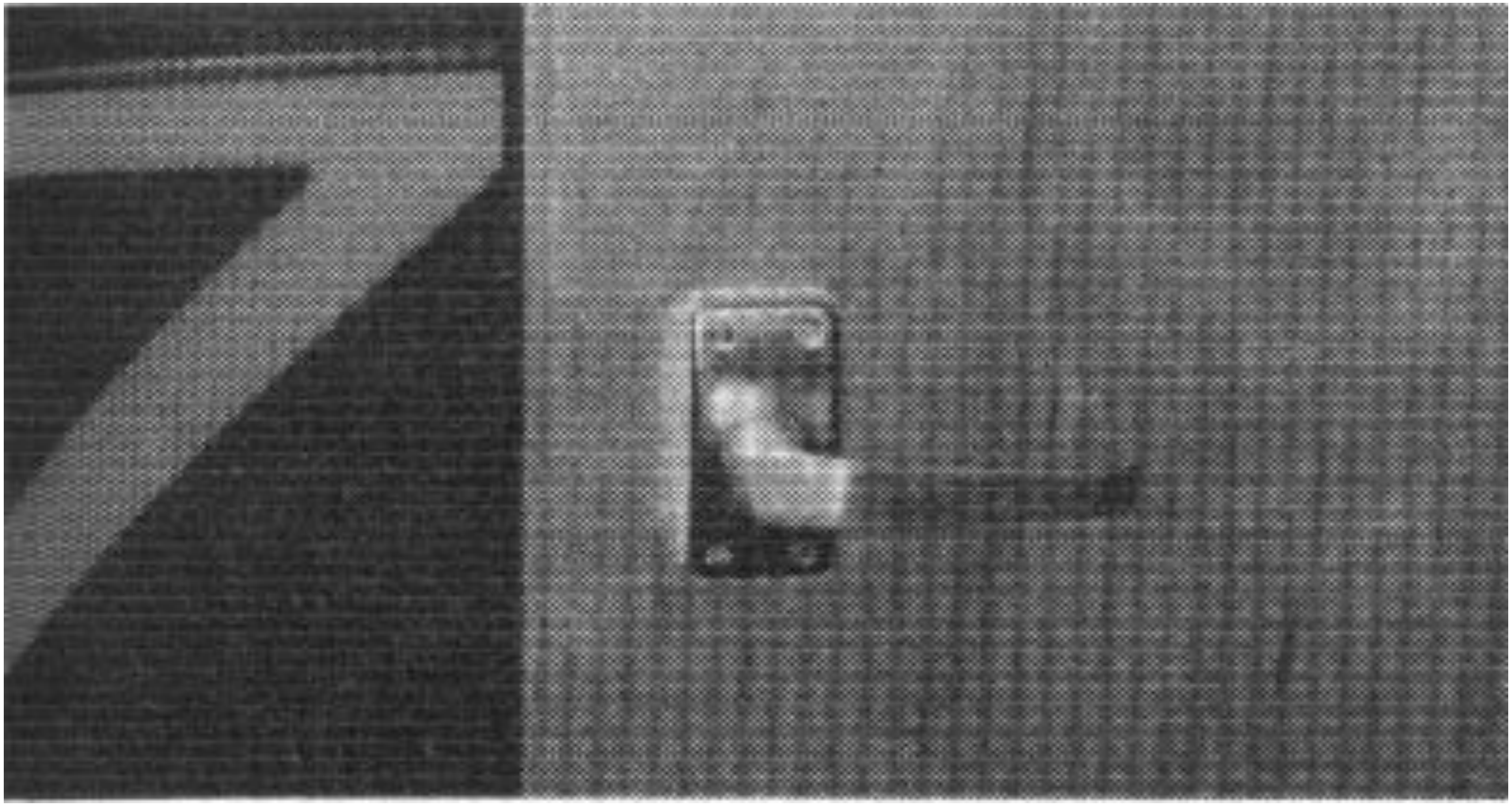
"If a door handle needs a sign,
then its design is faulty"
D. Norman,
The Psychology of Everyday Things



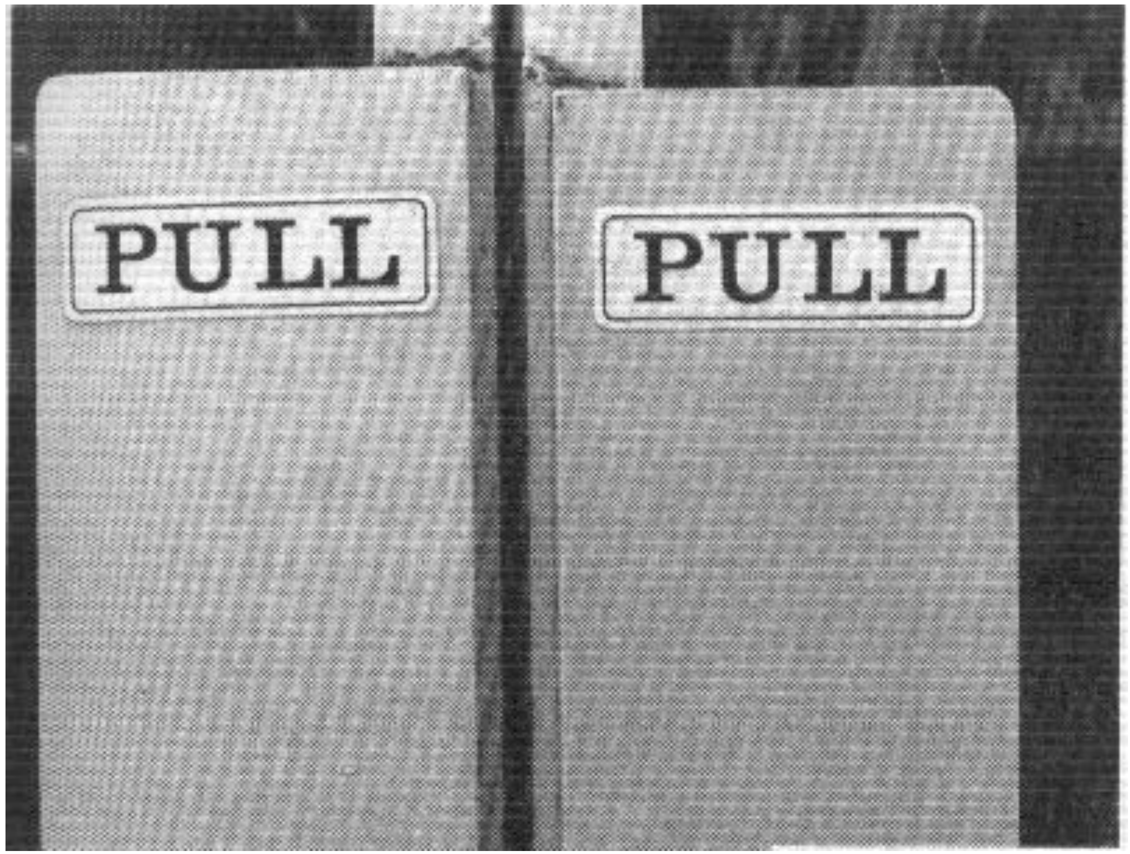
Norman on doors



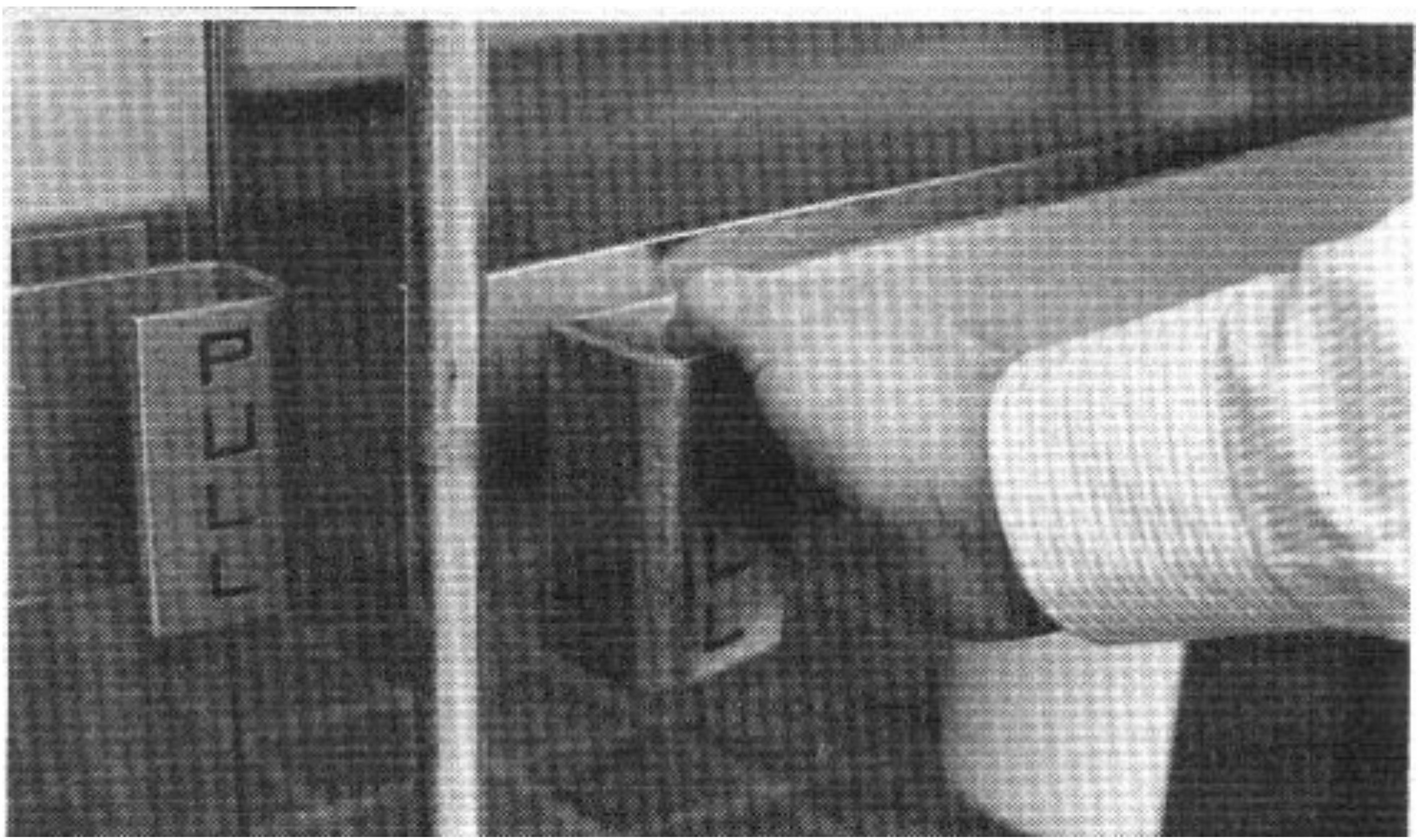
Norman on doors



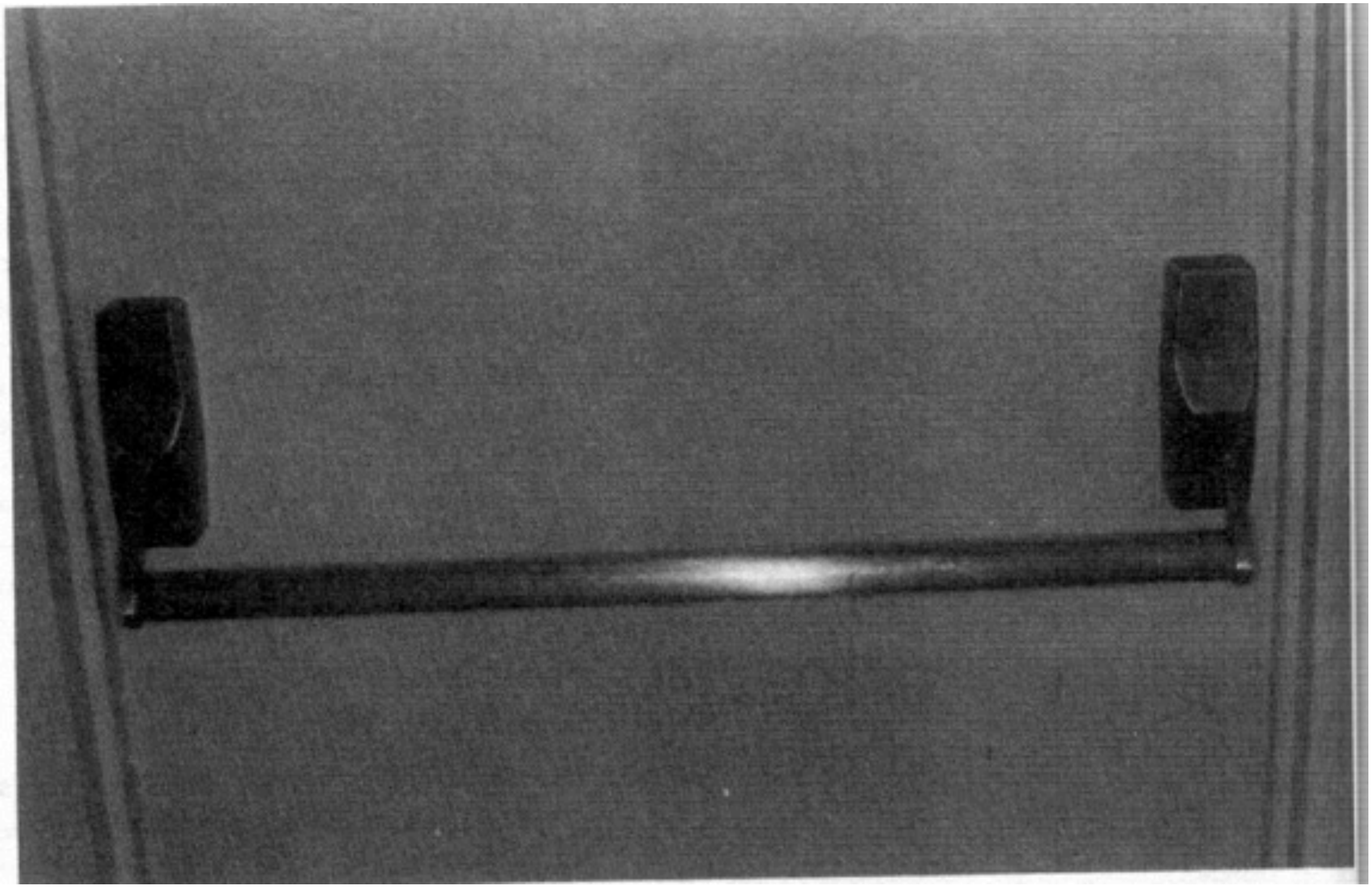
Norman on doors



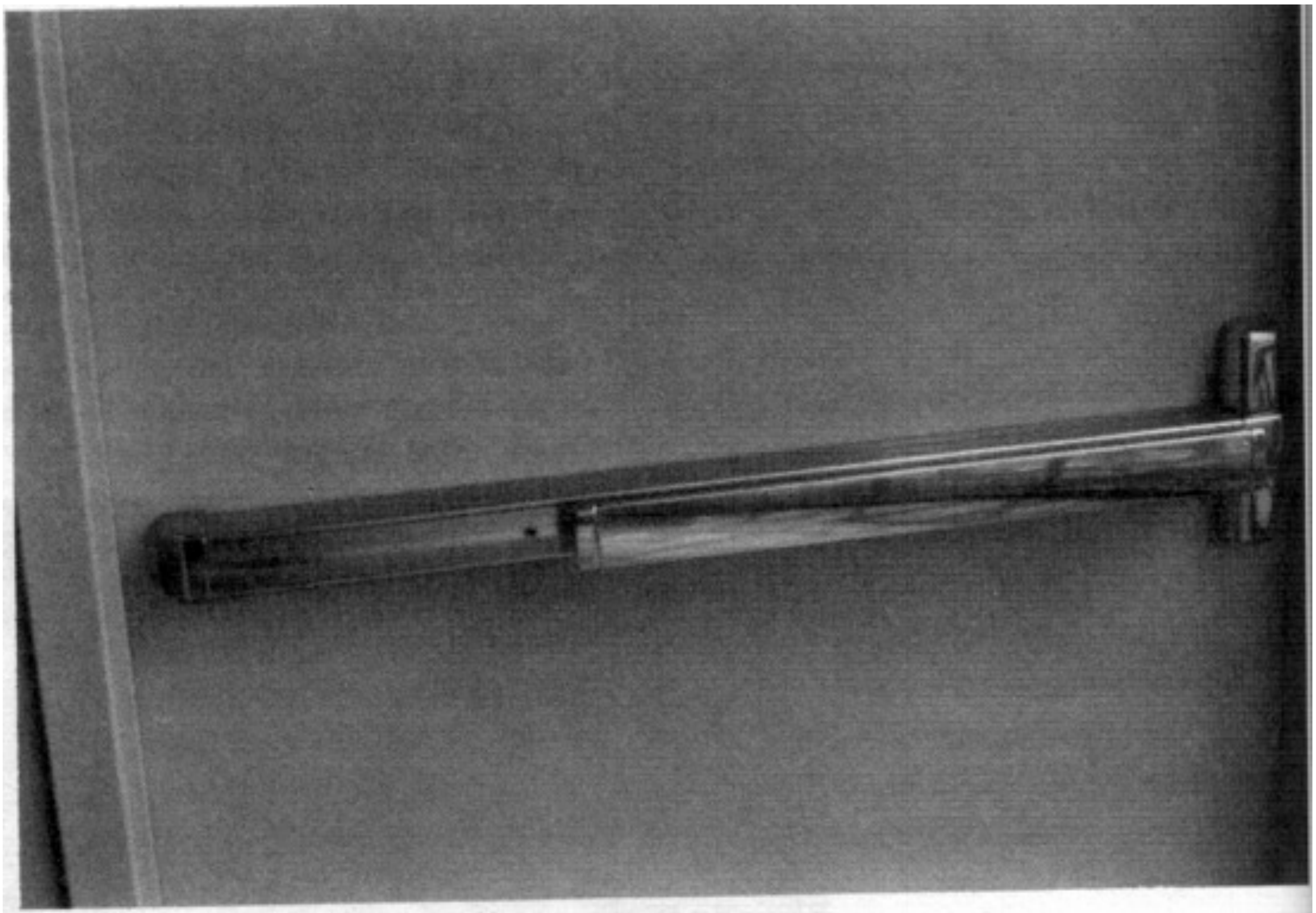
Norman on doors



Norman on doors



Norman on doors



Affordances & HCI

Doors are ‘surface artifacts’: what you can perceive is all that exists (though bad design might confuse these properties)

- E.g. physical file – can see if open, size, type of content

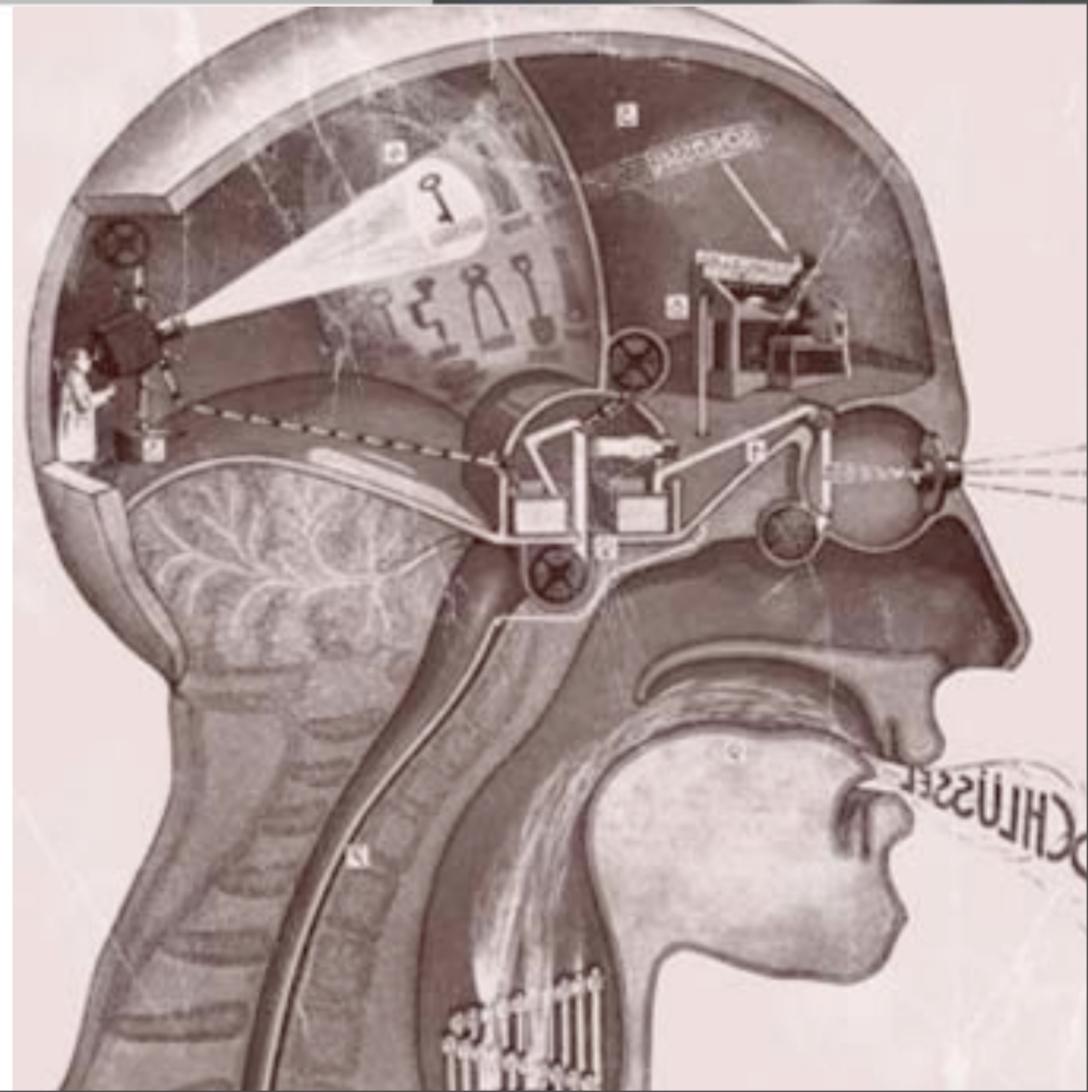
Computers are `internal artifacts’: they have complex internal states that determine their function but are not visible

This information needs to be transformed into a surface representation for the user:

- Opportunity: can choose the representation best suited to the user without the physical constraints of surface artifacts
- Problem: it is up to the designer to decide what will be visible; and this requires expert knowledge of both the artifact and the user
- E.g. what might the user need or want to know about the computer file?

Human capabilities: Memory and learning

Dr Mark Wright



Summary:

- Long term memory
  - organised as networks, schemas, frames
- Short term memory
  - what are the limits?
- Learning rates
- Learnability
- Affordances