

# The Psychology of Everyday Things

We use a variety of devices, instruments, computer programs, etc., everyday.

These include:

- digital watches
- mobile phones
- doors
- dvd recorders
- microwaves
- voicemail systems
- 

Some are easy to use others are difficult and frustrating to use.

## Why are some objects frustrating to use?

Due to poor design:

- they provide no clues or false clues to their operation
- they trap the user
- they thwart the normal process of interpretation and understanding

**Poor Design** leads to Frustration

- leads to Confusion
- leads to Error

On the other hand, **well-designed objects** are

- easy to understand
- easy to interpret
- use visible cues to their operation

## **To Become Better Designers, we can:**

- make use of affordances
- make use of constraints
- provide a good conceptual model
- make things visible
- use a good mapping - a natural one if possible
- provide feedback
- keep the number of features, actions and controls balanced

## **Make Use of Affordances**

**Affordances** refer to the perceived and actual properties of an object, which help the user determine how to use or interact with it

When affordances are used,  
the user knows what to do just by looking at it

Complex things might require some explanation,  
but simple things do not

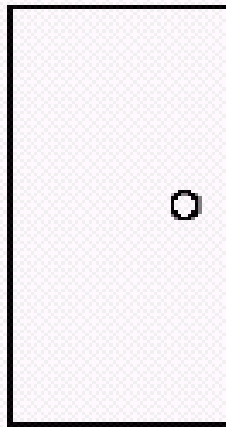
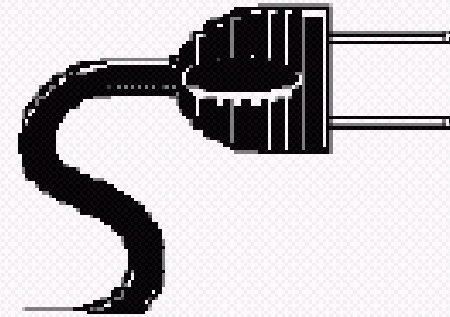
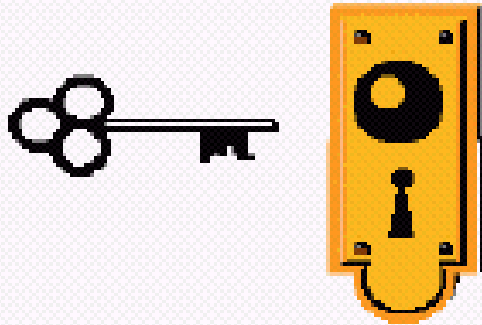
### **Principle of Affordance:**

It should be obvious how a control is used.

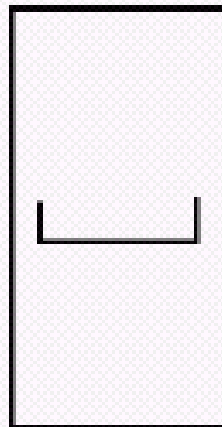
## **Examples of good affordances:**

- plates on doors - push
- knobs - turn
- slots - insert
- buttons – for pushing
- chairs - for sitting
- glass - break
- paper - write on, fold

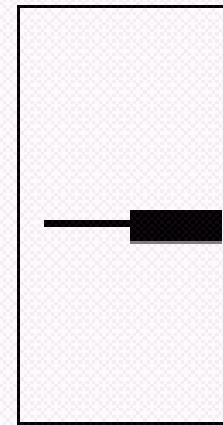
## Some Examples of Visual Affordances:



Push or pull?



Which side?



Can only push,  
side to push clearly visible

## **Make Use of Constraints**

**Constraints** restrict the allowed behavior or interaction with an object

### **Physical constraints**

The physical properties of an object constrain the possible operations:

- the order in which parts can go together

- the ways in which an object can be picked up, moved, manipulated

  - examples : scissors, doors, drawers, zippers, etc.

## **Cultural constraints**

Constraints which have evolved through artificial conventions that govern acceptable social behavior.

These cultural conventions have to be learned, but once learned apply to a wide variety of circumstances:

- tighten screws by turning clockwise
- loosen screws by turning anti-clockwise
- desktop metaphor used in GUIs

## Conceptual Models

Conceptual models are mental models, models that people have of themselves, others, the environment and the things with which they interact.

People form mental models

- from experience
- from training and instruction

The mental model of a device is formed by the interpretation of its perceived actions and its visible structure.

Upon seeing an object or a device, the user forms a mental model of how the device or object operates.

Users use the model to simulate the operation of a device or object.

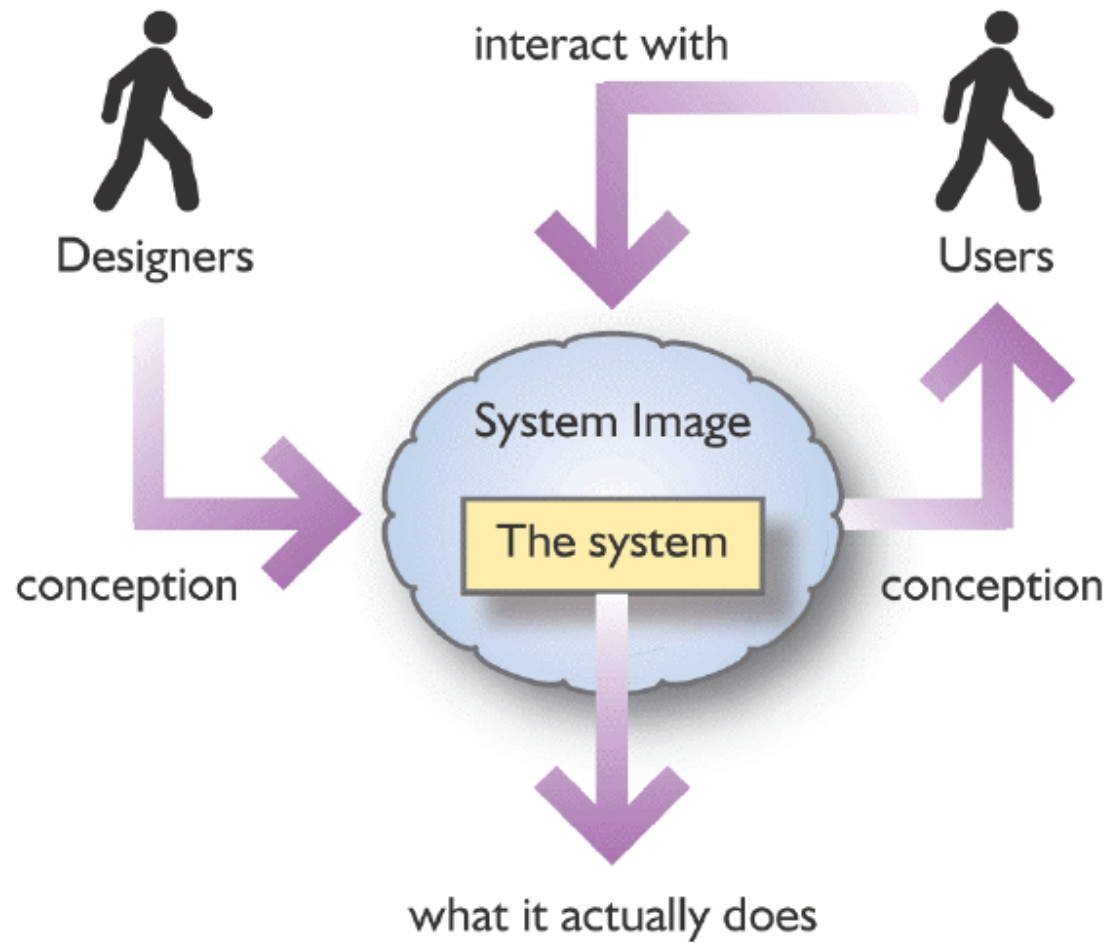
A **good conceptual model** allows users

- to predict the effect of their actions
- to understand the relationship between the controls of a device and the outcome

A **poor conceptual model**

- forces users to operate by rote, blindly
- makes it difficult to determine the effects of actions
- makes it difficult to figure out what to do in novel situations

## Three kinds of Conceptual Model



## **DESIGN MODEL**

The designer's conceptual model

## **SYSTEM IMAGE**

The visible structure of an object. It results from the physical structure that has been built, including documentation, instruction and labels.

## **USER MODEL**

The model the user develops through interaction with the system

Note that the designer would like:

**DESIGN MODEL = USER MODEL**

But the user never sees the designer's model, only the **SYSTEM IMAGE**.

# Conceptual Models for Computers

- Computers are much more complex than physical objects
- As a general purpose machine, a computer does not have a conceptual model
- Computers are used for many different purposes and run many different applications
- No one conceptual model is possible
- Designers must come up with a good underlying model that helps the user understand how to use the system

## The Use of Metaphor

Often designers employ **metaphors** to help the user form a suitable mental model.

Metaphors can be used to develop interfaces for applications.

APPLICATION AREA =====	METAPHOR =====	FAMILIAR KNOWLEDGE =====
Operating systems	The desktop	Office tasks
Spreadsheets	Ledger sheet	Columnar tables
Object-oriented environments	Physical world	Real-world behavior
Hypertext	Note cards	Flexible organization of structured text
Learning environments	Travel	Tours, guides, navigation
File storage	Files	Categorizing objects in in terms of urgency, projects, etc
Multimedia environments	Rooms (each associated with a different medium/task)	Spatial structure of buildings
Computer supported cooperative work	Multi-agents	Travel agents, butlers, and other serving roles

Metaphor can also be applied through analogy even if the metaphor is not concrete in the user interface (e.g., use a word processor like a typewriter).

When you choose a metaphor, you deliberately try to exploit the user's existing knowledge of the physical analogue.

**But beware of:**

- using metaphors that do not behave as the user might anticipate
- using metaphors too rigidly and inefficiently
- using metaphors that relate to objects outside the user's experience

## Make Things Visible

Visibility is an important principle of design and is used to:

- make the operation of a device understandable
- act as a reminder of what can and cannot be done
- make the state of the system clear

Visibility is achieved by

- making the correct parts or controls visible
- conveying the correct message

When the number of possible actions exceeds the number of controls, some functions become invisible, resulting in complexity.

Good visibility leads to objects/devices that are:

- easier to understand
- easier to use
- quick to learn
- easier to remember

How to make things visible:

- employ natural signals
- use good mappings (see next slide)
- good placement of controls

### **Principle of Visibility:**

It should be obvious what a control is used for.

## **Make Good Use of Mappings**

A **mapping** is the relationship between two things

In the case of HCI, a mapping is the relationship between the controls in an interface and their intended function.

The easier a mapping is to learn and remember, the easier a device will be to use.

## Natural mappings

Natural mappings take advantage of physical analogies and cultural standards

Natural mappings come from

- spatial analogy - press up button to make elevator go up
- perception - louder means greater

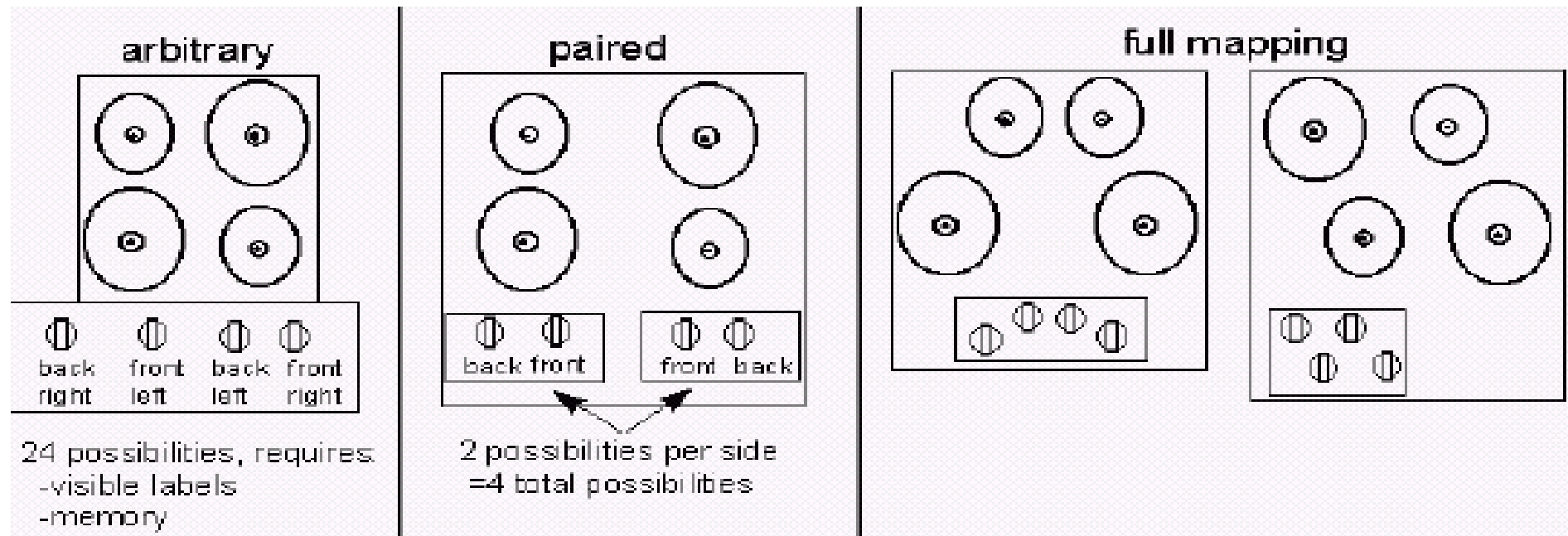
Some relationships do not have a natural relationship

e.g., pitch/hue/taste

does higher pitch mean more or less of something?

A device is easy to use when there is visibility to the set of possible actions and where the controls and displays exploit natural mappings.

# Example Mappings



## **Provide Feedback**

**Feedback** is the act of sending information back to the user about what has actually happened as the result of his actions or about the state of the system.

Feedback can be presented visually or aurally.

Good example:

simple push-button phones

Bad example:

complicated, multi-feature, modern telephones

### **Principle of Feedback:**

It should be obvious when a control has been used.

## **Manage Complexity**

Today's devices and computer systems are commonly developed with many, many, features.

However, the increase in controls and features makes it more difficult

- to make all the controls visible

which makes it harder for the user to

- understand the device
- learn how to use it, and
- memorize functions

Keep the number of features, actions and controls balanced.

## **Making mistakes**

### **Falsely Blaming Yourself**

- We often make mistakes when using devices, software, etc.
- For a complex task, this is reasonable and expected.
- For a simple task, or one that appears so,

**WE BLAME OURSELVES**

## **Norman's thoughts about Errors**

If an error is possible, someone will make it.

Designers should

- assume all possible errors will occur
- minimize the chance of errors
- minimize the effects of errors when they do occur
- make it easy for users to detect errors
- make it possible to reverse the effects of an error

## People Will Make Errors

Complex devices and software will always require some instruction.

Someone using them without reading the manual (very common among computer users) should be expected to make errors and to be confused.

As designers, we should design for error by:

- minimizing the possibility for error
- making errors as “cost-free” as possible

## **Where To Place the Blame?**

When something goes wrong with technology who do we blame?

Simple tasks (or tasks we think others can do) ...

- **we blame ourselves**

In more complex cases, we have a tendency to search for the cause of an event. We find a perceived casual relationship between the thing being blamed and the result.

- **we blame others**
- **we blame the environment**

We blame problems of other people on their personalities.

But ... really ...

- a faulty mental model is at work!

## Misconceptions of Everyday Life

Misconceptions can be due to “naïve” or “folk” understandings  
but are mostly due to **faulty conceptual models** we form.

Models are based on whatever knowledge we have, real or imaginary,  
naive or sophisticated.

Models are constructed out of fragmentary evidence, with a poor  
understanding of what is happening.

People do not have erroneous theories...

Instead we form mental models to explain what we observe.

With the lack of complete information, we let our imaginations run free  
as long as the model accounts for the faults as we **perceive** them.

## Norman's Model of Action

*"The basic idea is simple. To get something done, you have to start with some notion of what is wanted—the goal that is to be achieved. Then, you have to do something to the world, that is, take action to move yourself or manipulate someone or something. Finally, you check to see that your goal was made. So there are four different things to consider: the goal, what is done to the world, the world itself, and the check of the world. The action itself has two major aspects: doing something and checking. Call these execution and evaluation."*

[Norman]

# Norman's Seven Stages of Action

## 1. Forming the Goal

Something to be achieved. Can be stated in a very imprecise way; e.g., "make a nice meal".

## EXECUTION

## 2. Forming the Intention

Goals must be transformed into intentions, i.e., specific statements of what has to be done to satisfy the goal; e.g., "Make a chicken casserole using a can of prepared sauce."

## 3. Specifying an Action Sequence

What is to be done to the World. The precise sequence of operators that must be performed to effect the intention; e.g., "Defrost frozen chicken, open can, ..."

## 4. Executing an Action

Actually doing something. Putting the action sequence into effect on the world; e.g., actually opening the can.

# **EVALUATION**

## **5. Perceiving the State of the World**

Perceiving what has actually happened; e.g., the experience of smell, taste and look of the prepared meal.

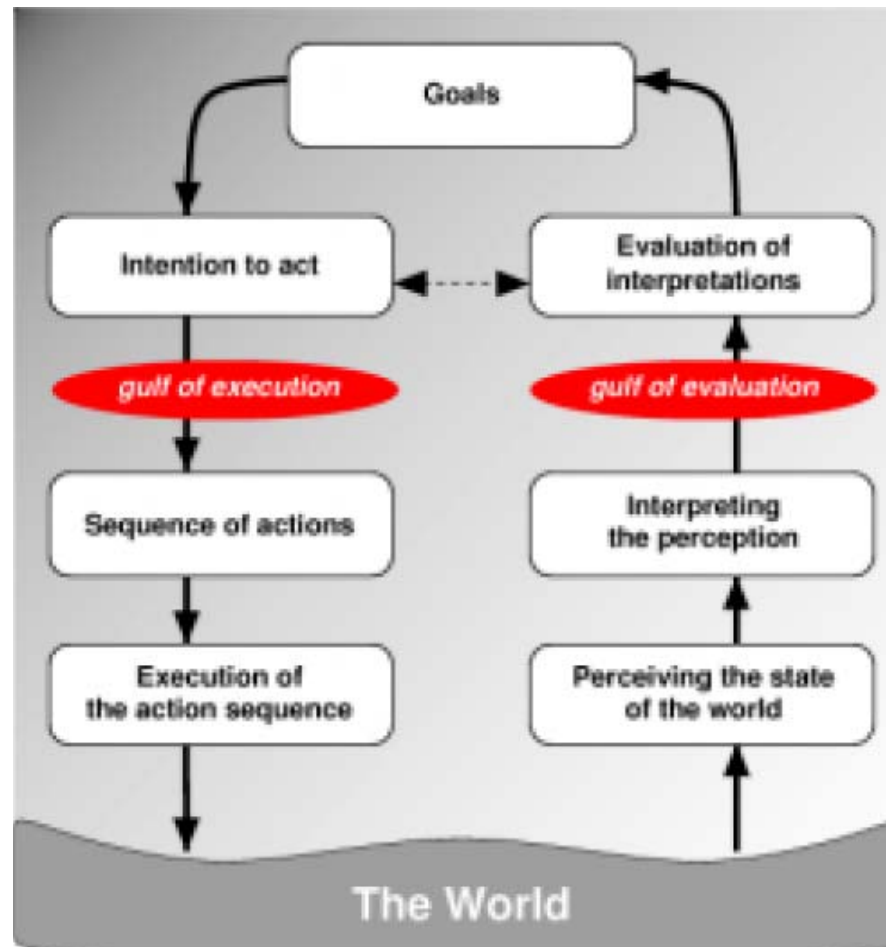
## **6. Interpreting the State of the World**

Trying to make sense of the perceptions available; e.g., Putting those perceptions together to present the sensory experience of a chicken casserole.

## **7. Evaluating the Outcome**

Comparing what happened with what was wanted; e.g., did the chicken casserole match up to the requirement of 'a nice meal'?

# The Stages of User Activities When Performing a Task



## **The Seven Stages as an Approximate Model**

Stages are not discrete entities.

Not all stages are required for every goal.

Most goals are not satisfied by a single action.

Numerous sequences.

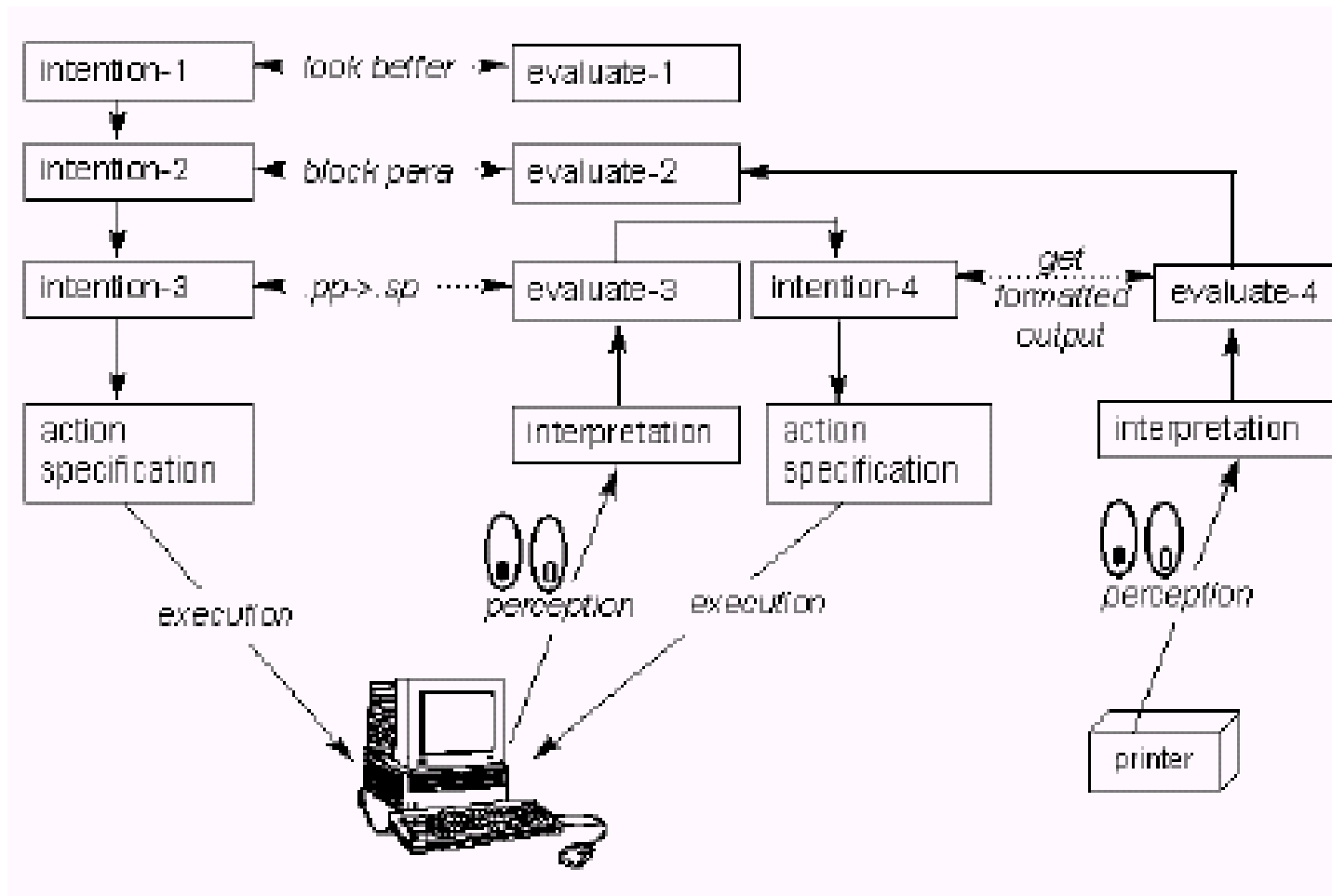
May span seconds or minutes or hours or days.

Continuous feedback

- results may spawn other goals and other actions
- goals lead to sub-goals
- intentions lead to sub-intentions

In a large activity, intermediate goals can be forgotten, discarded or reformulated.

## A Typical Task: Improving a Business Letter



## Opportunistic Actions

Humans do not plan everything.

We are spontaneous.

Goals are often ill-formed and vague.

We respond to events in the world.

We are **data-driven**: as events in the world around us unfold, we introduce new goals, which lead to new actions, as **opportunity** allows us.

## **The Gulf of Evaluation**

Reflects the amount of effort a person must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met.

Does the system provide a physical representation that can be directly perceived?

Can the person easily interpret representation in terms of his intentions and expectations?

When users can easily see that the system has done what they wanted it to do, the gulf of evaluation is insignificantly small.

When it is hard to tell what the system has done, the gulf of evaluation has appeared, and the more effort that is required to evaluate the system state, the larger that gulf has become.

## **The Gulf of Execution**

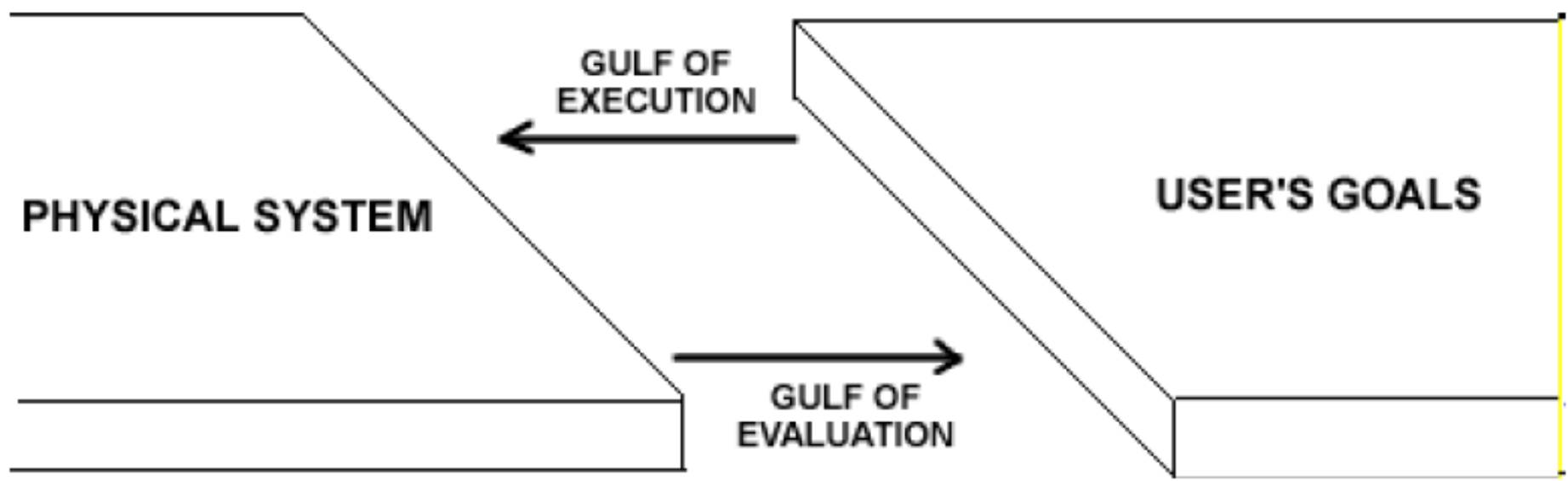
Refers to the difference between the user's goals and what the system allows them to do; how well the system enables a person to perform the intended actions directly without extra effort.

How well does the system allow the person to do the intended actions directly?

Do the actions provided by the system match those intended by the person?

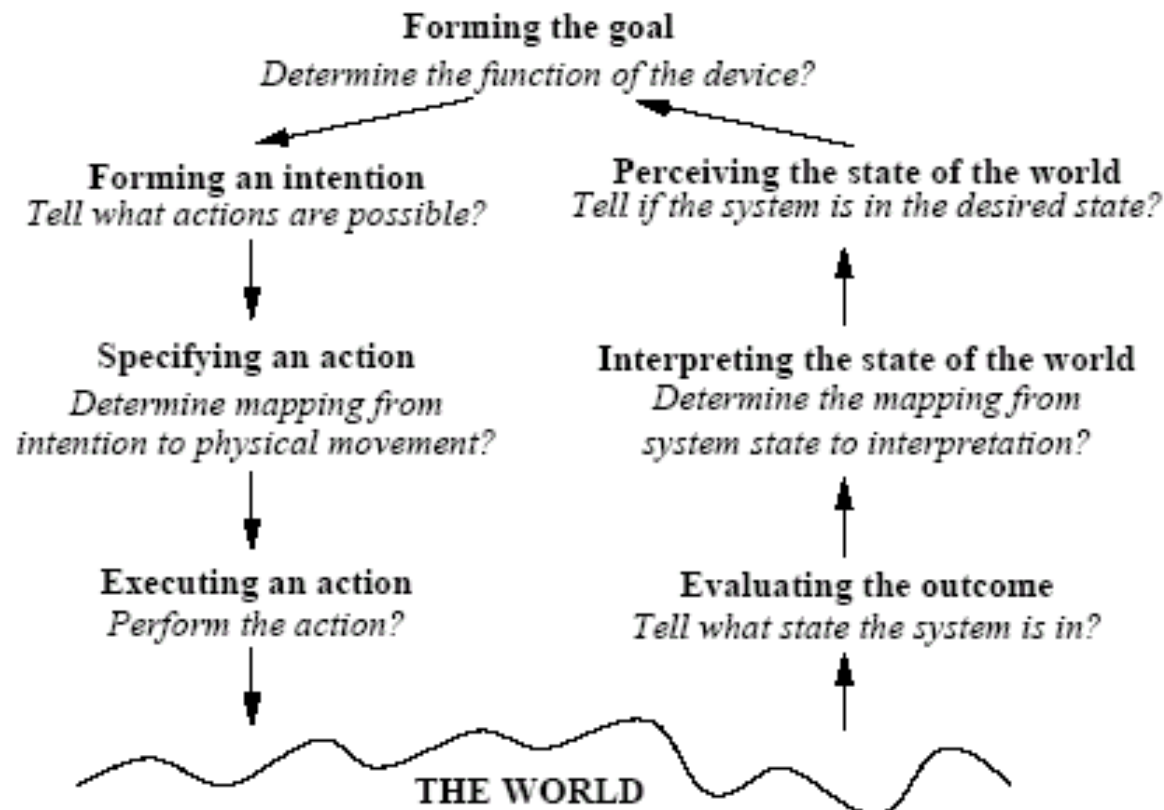
When users can easily see how to use the physical system or object to achieve their goals, the gulf of execution is small and insignificant.

It increases in size as the product or system requires more effort to use.

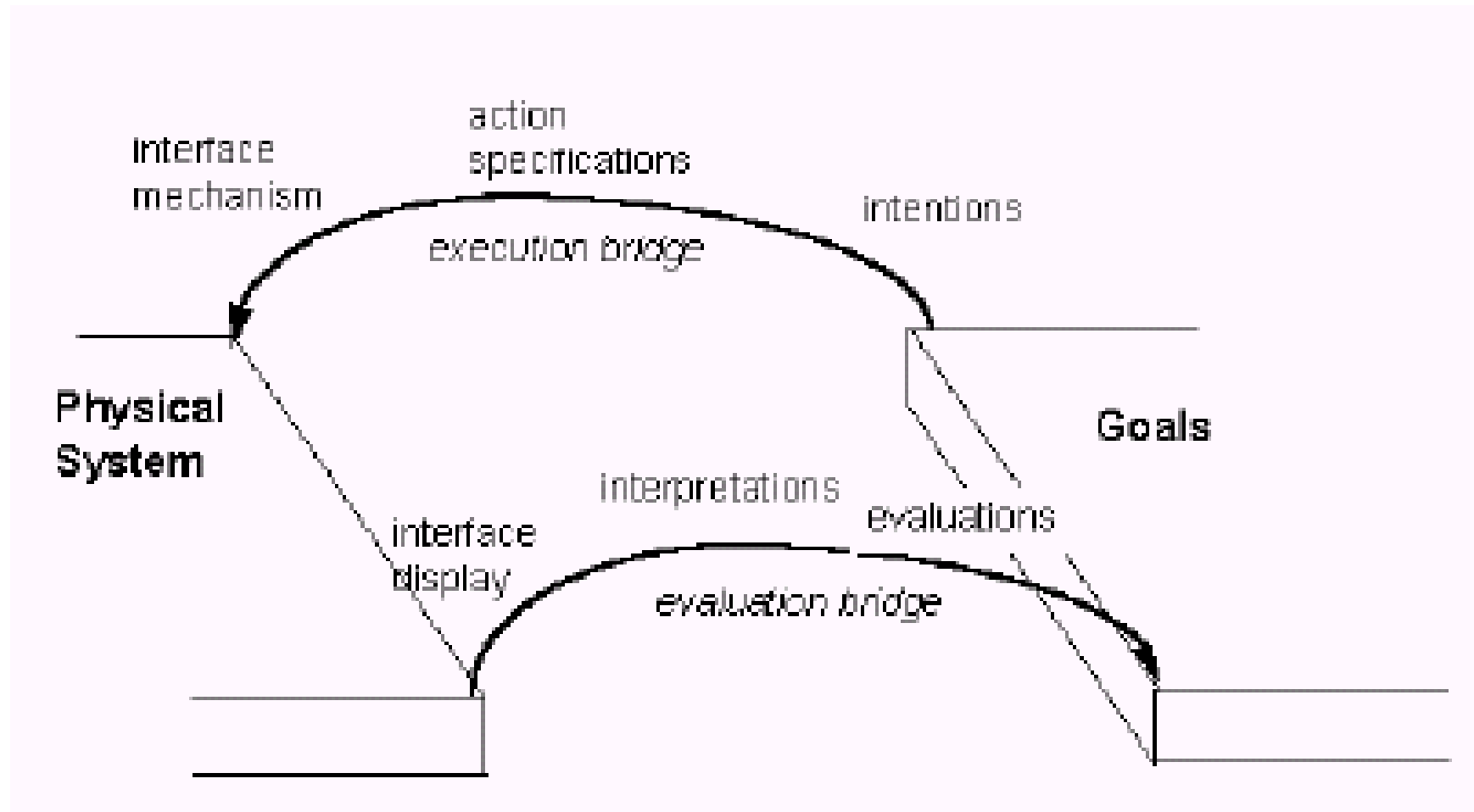


## The Seven Stages as a Design Aid

- Questions we can ask to ensure the gulfs are “bridged”:  
How easily can the user ...



# Bridging the Gulfs



## **This leads back to our Design Principles:**

### **Provide a good conceptual model**

- Coherent system image.
- Consistency in presentation of operations and results.
- Metaphor?

### **Make things visible**

- Is the state of the system easily visible.
- Can alternative actions be easily found.

### **Use controls with good (perceptual) affordances**

- Is it clear how the controls can be used?

Use a **good mapping** - a natural one, if possible, showing relationships between

- actions and results
- controls and their effects
- system state and what is visible

Provide **feedback**

- Continuous feedback of results and actions