

Design Cognition

CS 347

Michael Bernstein

Announcements

Quiz after lecture today — pencils!

Ubiquitous computing

Unit I

ubiquitous and tangible computing
input and output
activity sensing

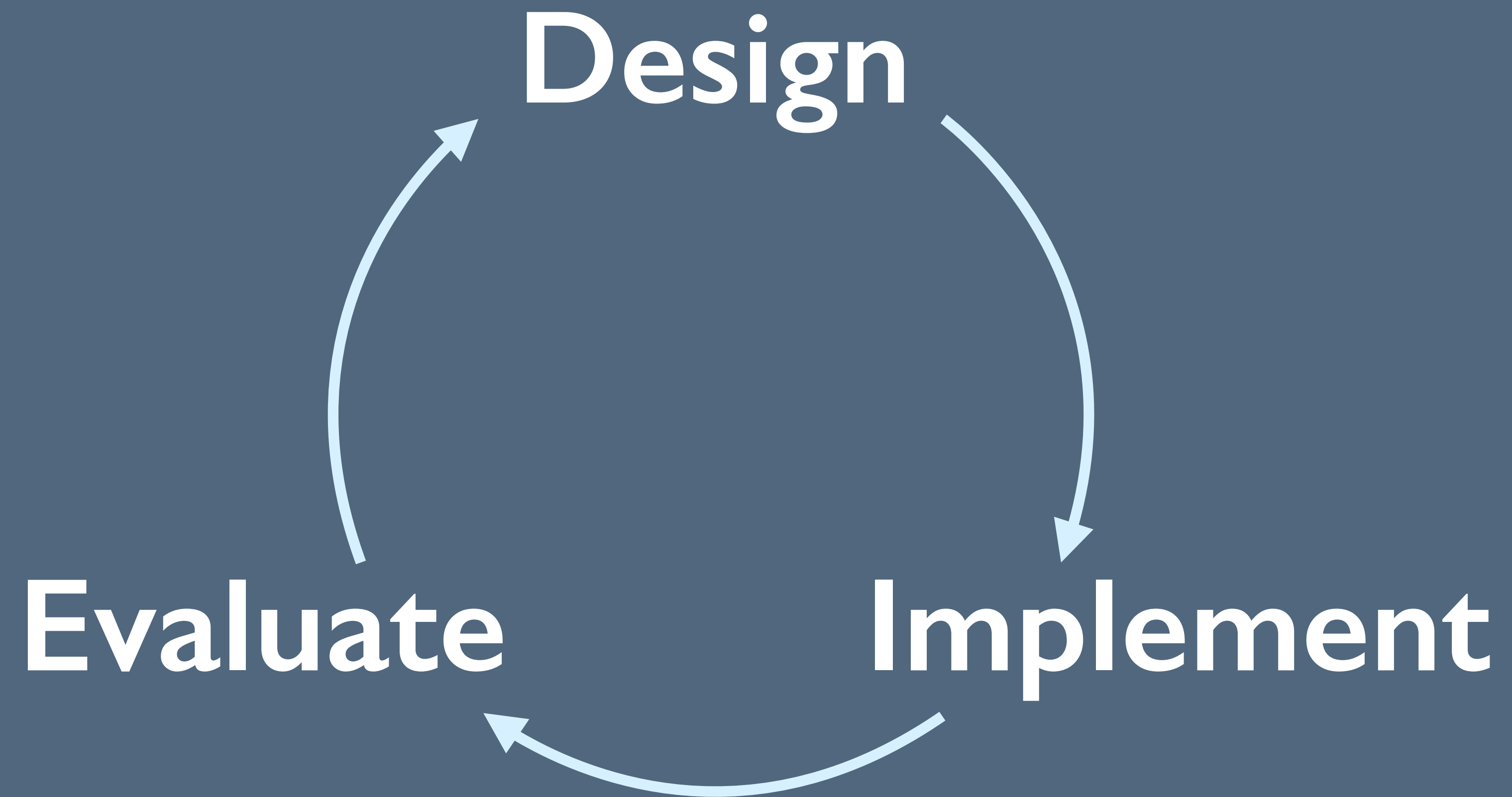
Design

Unit 2

design cognition

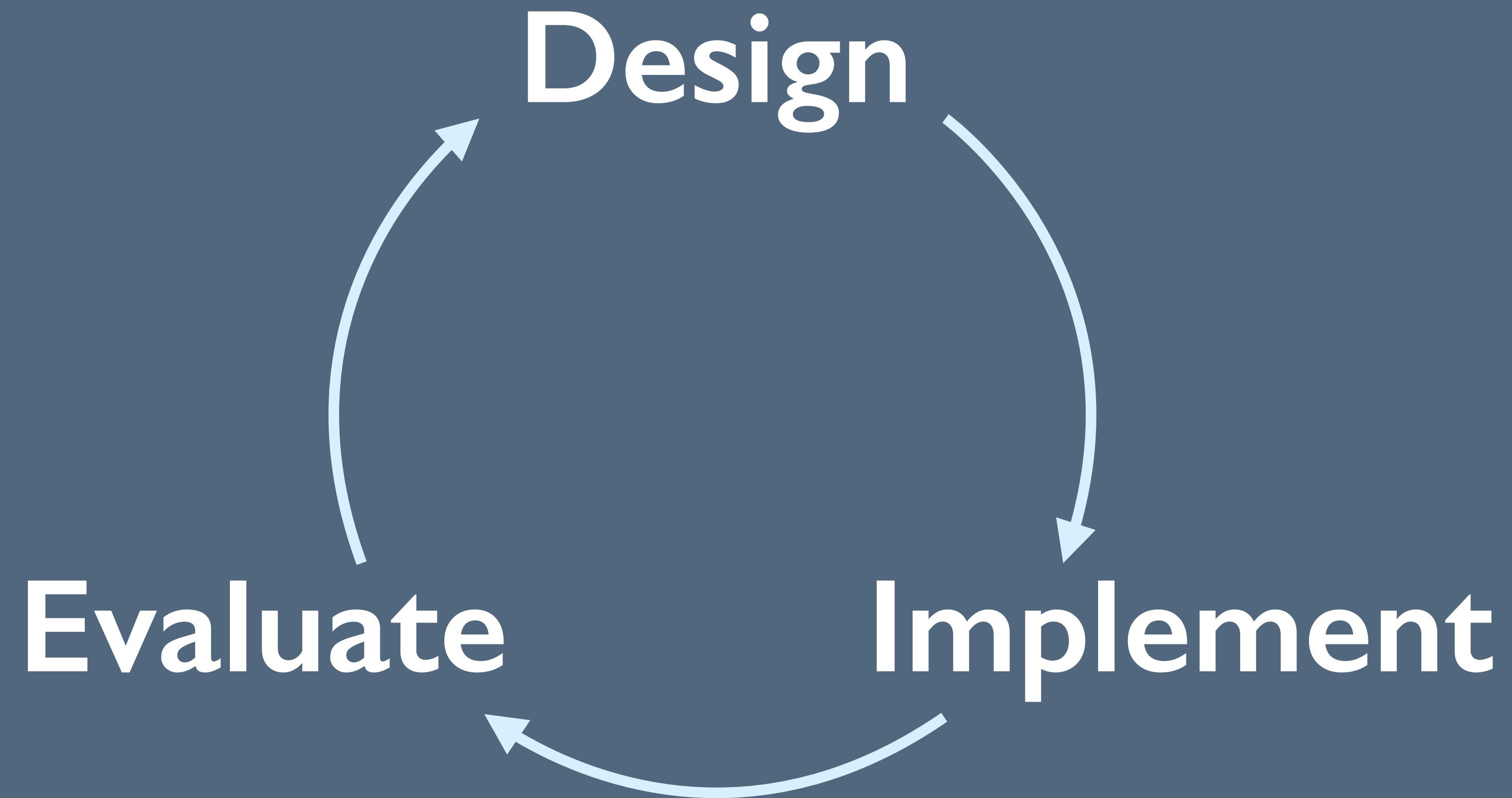
design process

design tools



Design is not a *static*
process.

It can be *studied*,
supported, and *improved*.



**How might we facilitate and empower
this process?**

Design

Brainstorming process
Early-stage design tools

Evaluate

Study strategies
Cognitive modeling

Implement

Programming tools
WYSIWYG design tools
Rapid prototyping tools

Goal of the design unit

Refocus from the **process** of design to the **principles** that guide that process

Shift from obsequious adherence to a single prescriptive design process to an understanding of what each part of the process is trying to achieve

Result: apply the **right processes at the right time**, and develop entirely **new process innovations**

Today

Design cognition: how our thinking shapes our design process, and how our thinking shapes others' reactions to our designs

Three major themes of design cognition:

Design fixation

Analogical transfer

} Why you don't get what you want out of the design process

Gulfs of execution and evaluation

} How to argue why a design is (or is not) better

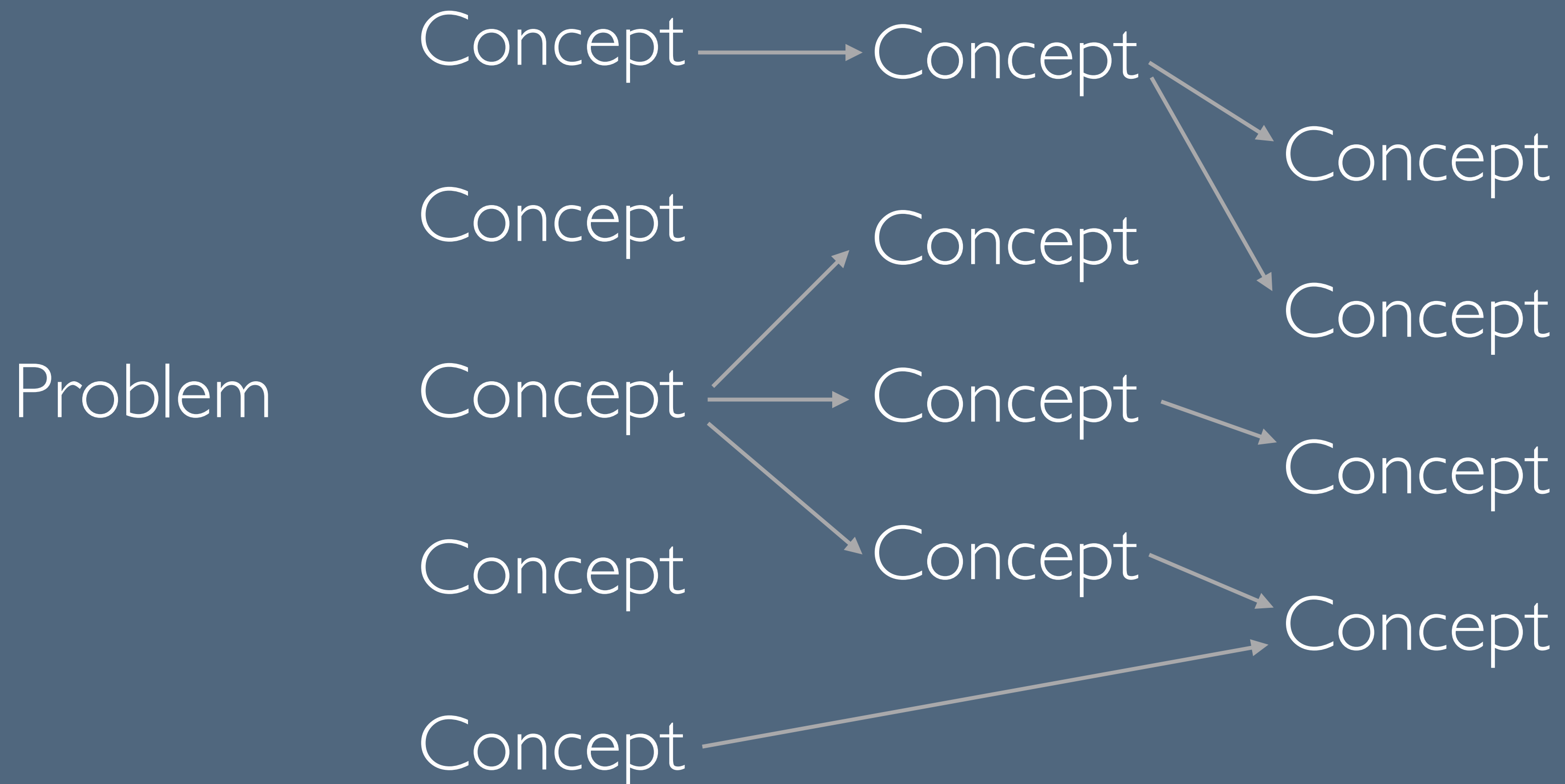
Why You Might Not Get What You Want

Part I

Design Fixation

Ideal: open-minded ideation

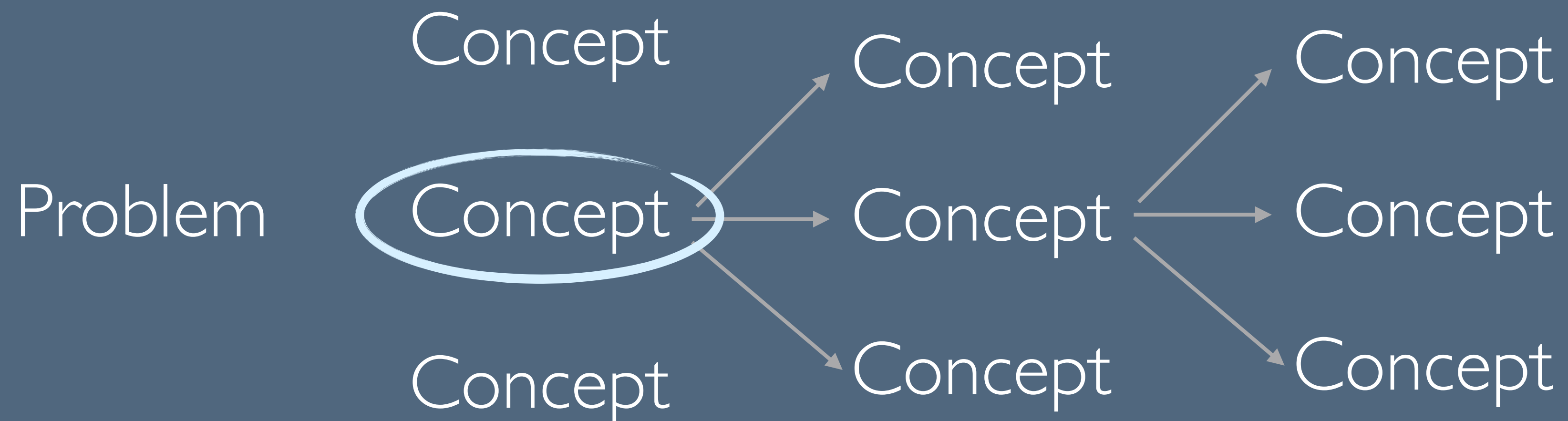
In theory, the ideation process identifies many ideas, both proximal and distal



Reality: not enough breadth

In practice, we often myopically stay near proximal concepts that we've used before or that are surface-level similar

Why?



“I always liked
this one anyway”

Design fixation

In cognitive psychology, **fixation** is when we introduce self-imposed barriers to problem solving [Maier 1931, Luchins 1942]

Design fixation is when we limit the breadth of our design process through adherence to a small set of concepts [Jansson and Smith 1991]

Design fixation takes hold both (1) unconsciously, when we're not aware, and also (2) consciously, even when we're aware that we're doing it.

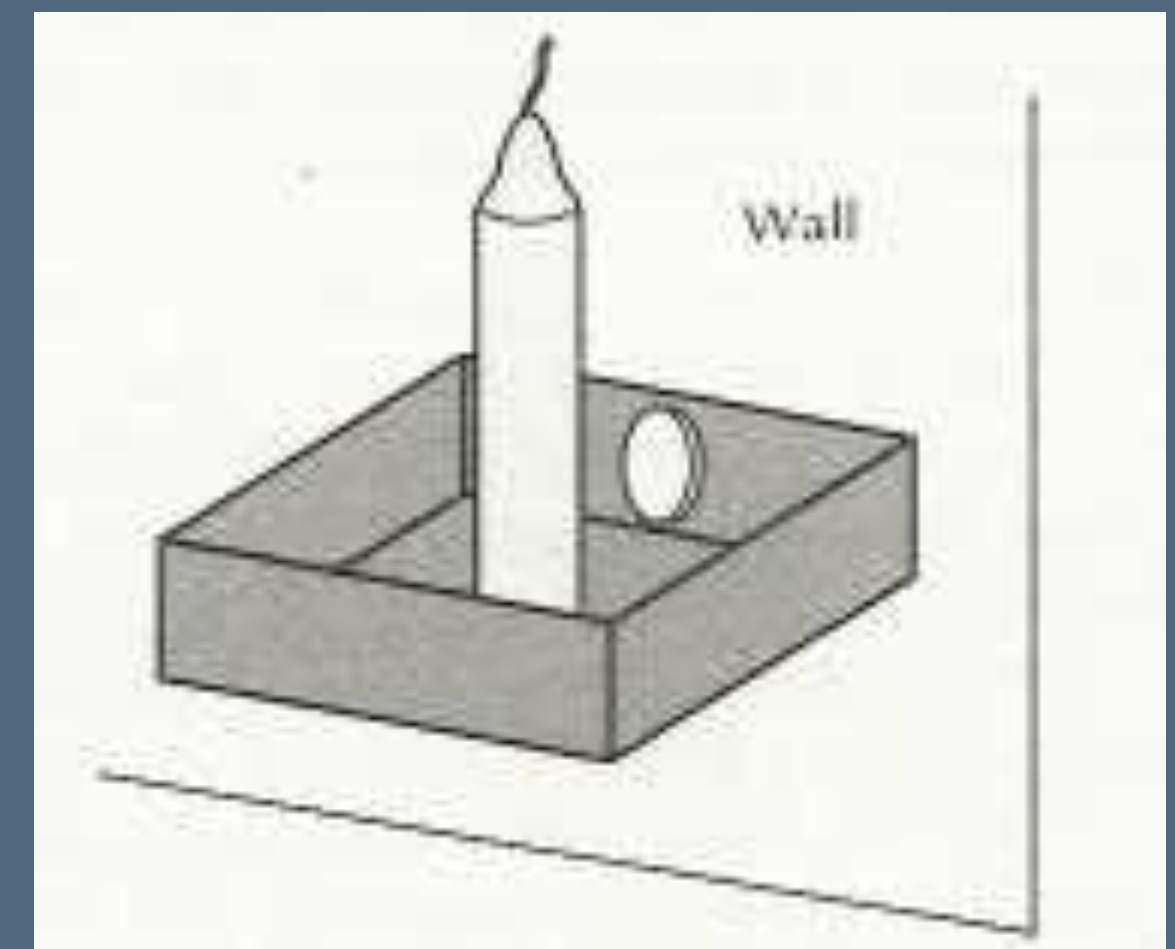
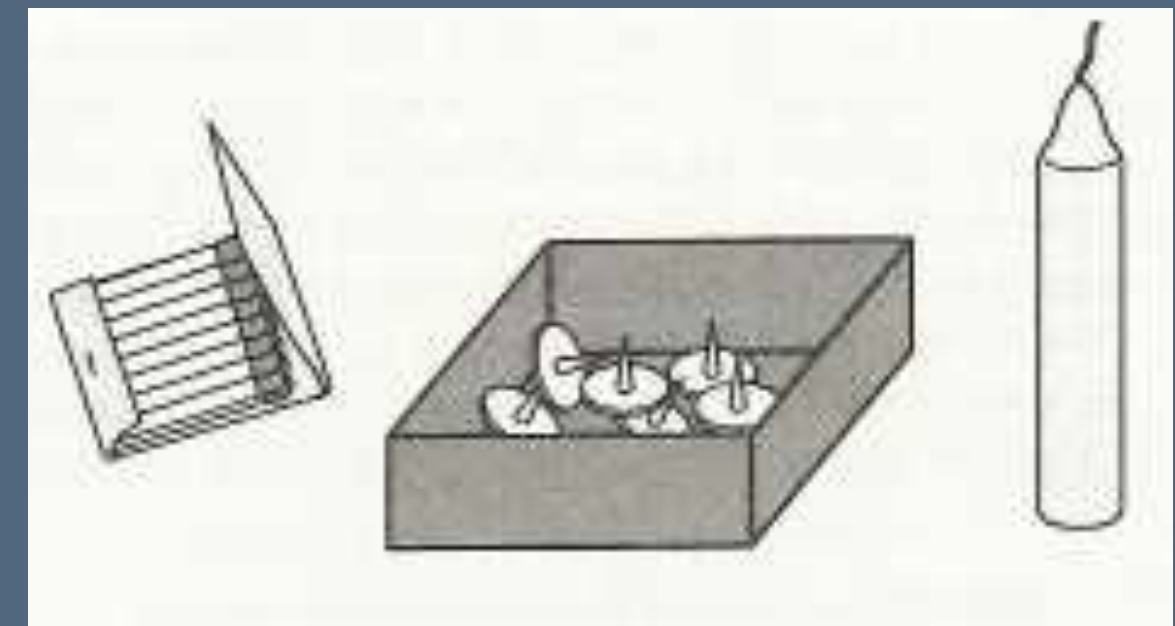
Classic example of fixation

[Duncker and Lees 1945]

Goal: attach a candle to a wall so that the candle won't drip on the floor. You can only use (1) a book of matches, (2) a box of thumbtacks.

Designers are trained to question assumptions, and to creatively recombine the tools at their disposal.

However, we are biased toward using objects only in the ways we've seen them used before.



Classic example of fixation 2

[Luchins 1942]

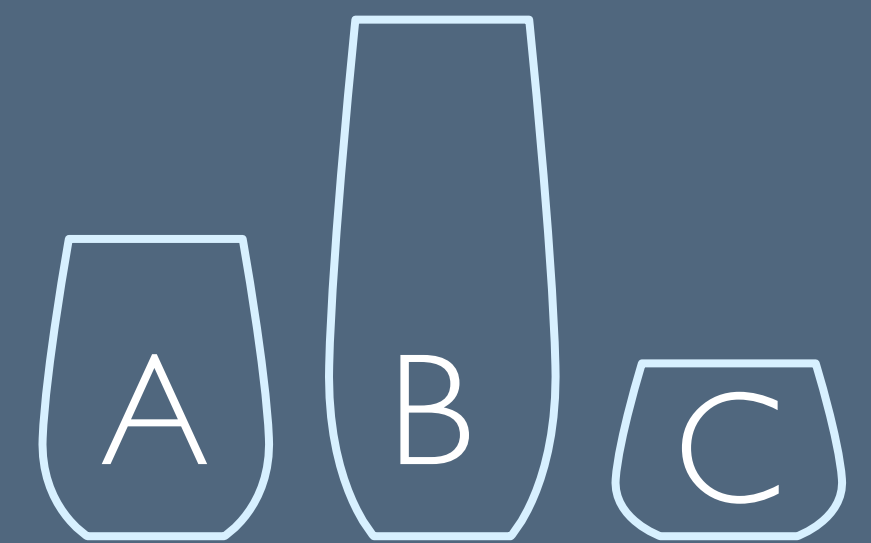
Goal: measure a specific amount of water with the jars

Method: participants were given **practice tasks** that could be solved via a nontrivial algorithm B-A-2C

The **test problem** could be solved via the nontrivial algorithm (B-A-2C) but also very simply (A+C).

70% still used the nontrivial algorithm.

The additional practice **should have made us better**. But, due to **fixation** on the approach we knew about, it made us **worse**.



[Professor Layton]

Even worse, we fall in love with our own ideas

The IKEA Effect [Norton, Mochon, Ariely 2012]: **we place high value on things that we helped create**

Experiment: One group of people build a piece of IKEA furniture, the control group get it pre-assembled. Both are asked how much they'd pay for the furniture.

Those who assembled their own box were willing to pay a 63% premium over those who received the same furniture pre-assembled

Ideally, showing other peoples' ideas should positively influence our ideation. Instead, **we tend to ignore others' ideas**—unless the person who came up with them joins our design team. [Choi and Thompson 2005]

The harms of design fixation

Fixation anchors us in a small subset of the design space, preventing us from identifying the best solution

Knowing that it's happening doesn't help us escape it

What does help us escape it?

Some designers and creative professionals practice **strategic forgetting**, where they intentionally don't capture ideas immediately, and trusting that good ideas will come back multiple times [Nicholas, Sterman, and Paulos 2022]

Another approach might be...[advances the slide]

Analogical transfer

Where do good ideas come from?

It's often easy to translate a solution from one problem to another problem if the **surface features** of the problems are similar.

Worked-out textbook solution

Test problem following the exact same format

But, major innovations are not such simple copy-pastes. They require mapping **deep features** between problems.

Fitting a solar array in 1/10th the size for takeoff

Origami

[Zirbel et al. 2013]

How do bacteria mutate?

Slot machines

[Murray 2016]

Analogical transfer

Transfer across these deep structures is referred to as **analogical transfer**, as in **transfer via analogy**

How? We abstract problems and solutions we've encountered into **schemas** that drop out surface features and facilitate comparison

Example

Problem: In embodied VR it is **difficult to work with objects that far away** from the user's avatar

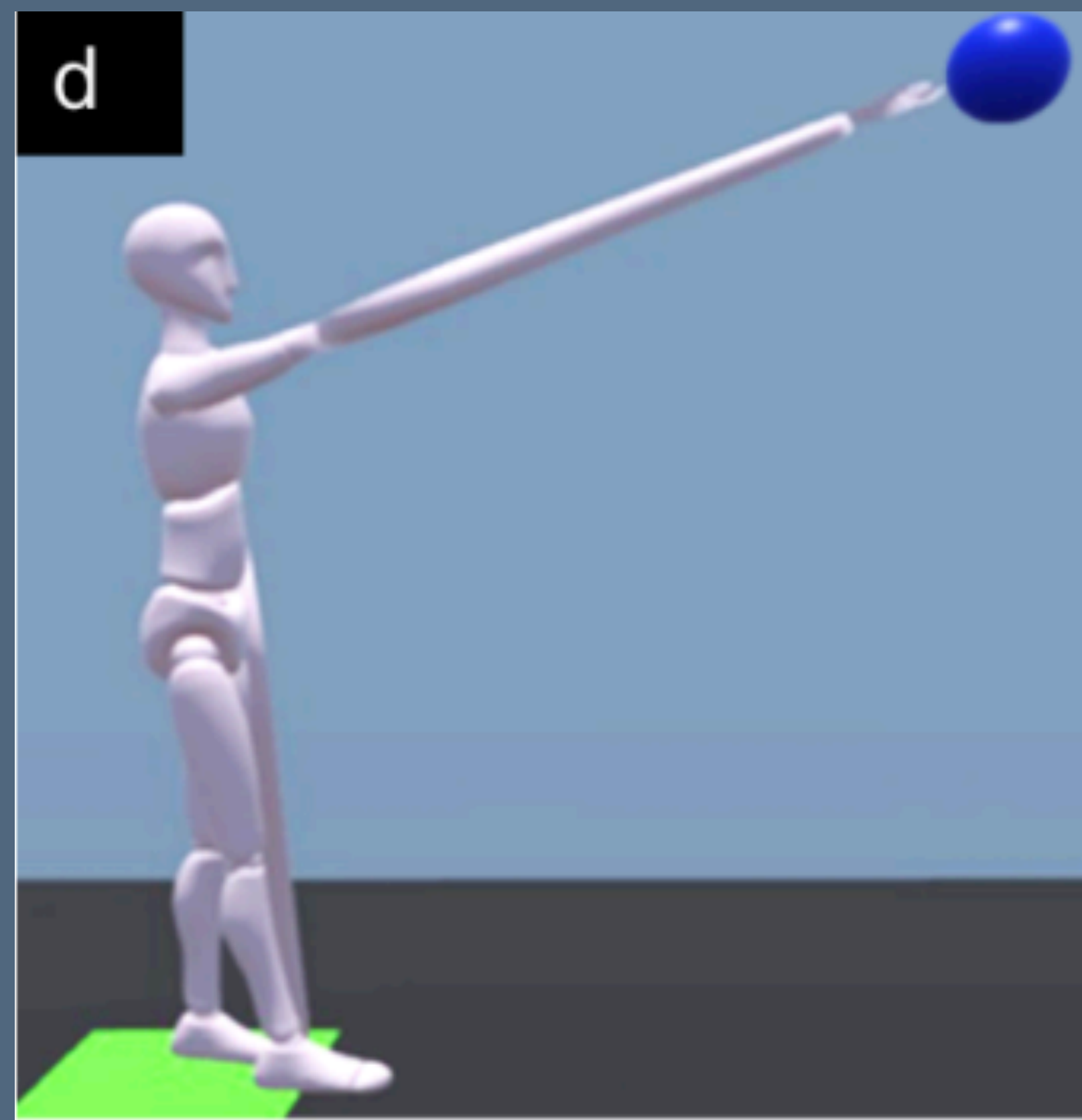
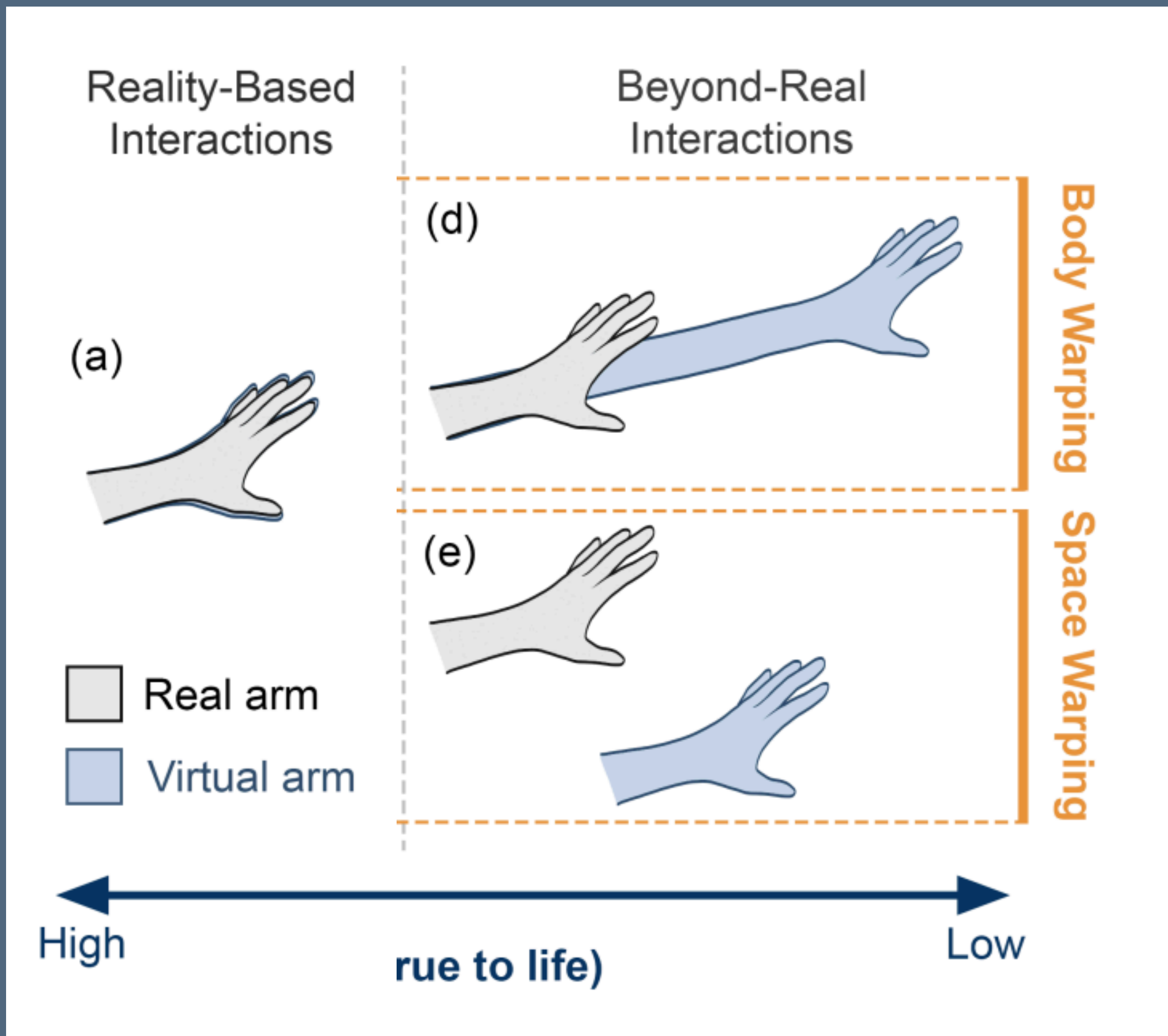
Surface feature transfer (boo): adapt laser pointer metaphor to VR

Deep feature, schema-based transfer (yay):

Social computing article: “Beyond Being There” [Hollan and Stornetta 1992] says to create collaborative experiences better than “being there”

Schema: Problem = adherence to reality is unsatisfying
Solution = stop trying to be realistic

Result: “Beyond Being Real” [Abtahi et al. 2022], create VR experiences that break from realistic self-representation



“Beyond Being Real” [Abtahi et al. 2022]

Barriers to analogical transfer

For analogical structure mapping to work, it requires that we create the correct schemas and retrieve based on those schemas

Unfortunately, we are biased against deep structural comparisons due to fixation: we tend to focus on surface features

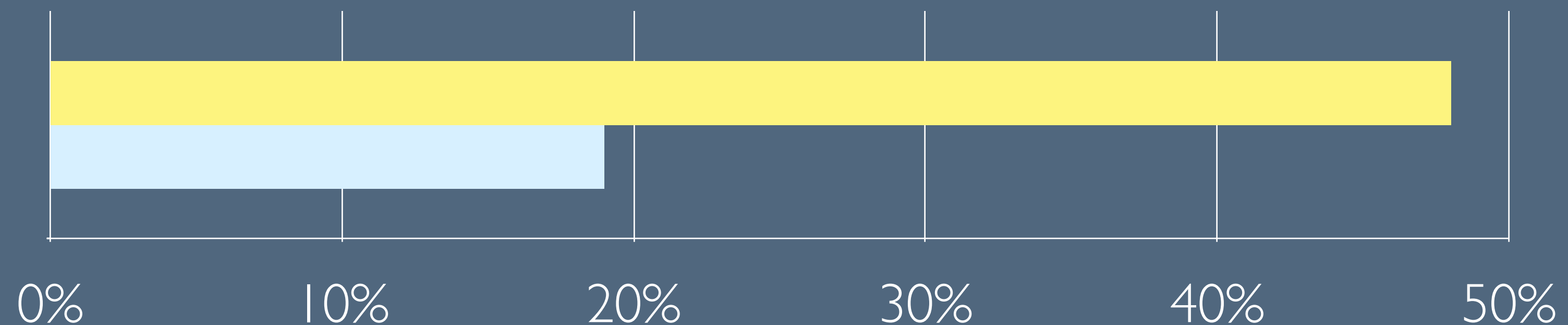
Study: When learning probability, participants were asked to recall earlier problems that were relevant. **80% of the recalled problems were based on surface similarities** (e.g., both about shopping lists) rather than the probability theory principles. [Ross 1984]

Without scaffolding, people don't identify deep features

Study: participants learning negotiation strategies

“Read these one at a time” vs “Compare these examples”

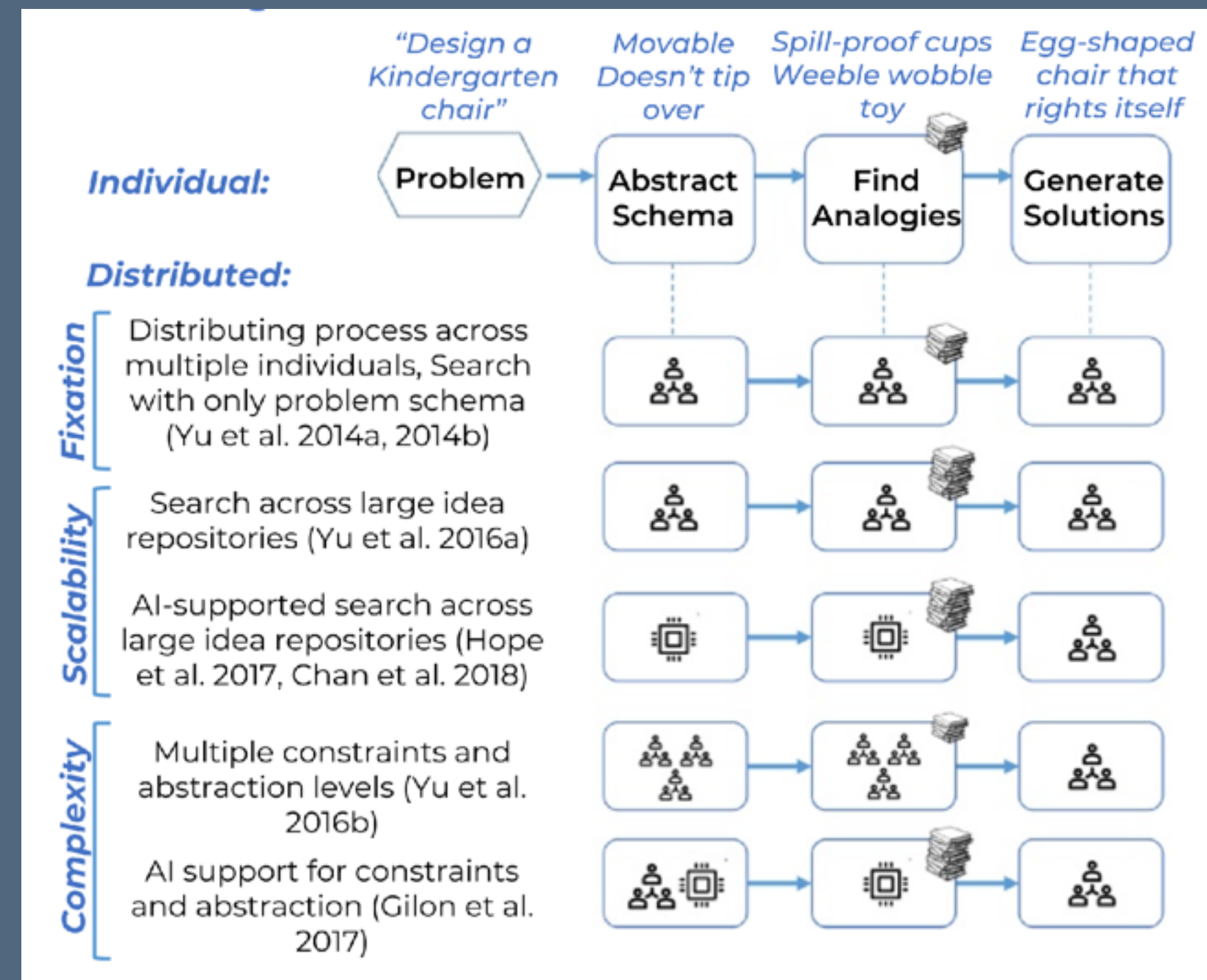
Measure: % of participants who correctly transfer the negotiation principle in the examples to a test case



Implications for design

Bad ideas are often a result of poor analogical transfer: applying surface features rather than deep features in retrieving relevant ideas

On the other hand, this raises opportunities: we can develop techniques to extract schemas at scale from existing ideas and aid application to new problems [Kittur et al. 2019]



How to Argue Why A Design Is (Or Is Not) Better

Part II

Gulfs of Evaluation and Execution

Goal: a cognitive account of why a design is poor

When people “don’t get it”, what’s actually happening?

When people do get it can we say more than : “It just feels **natural**”

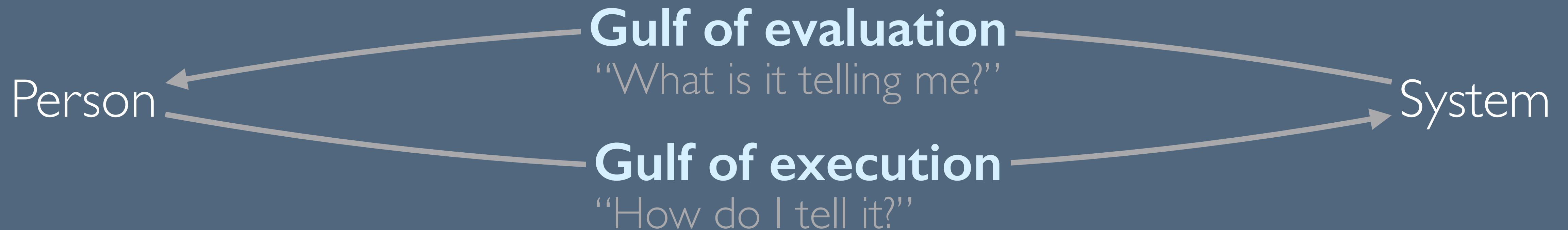


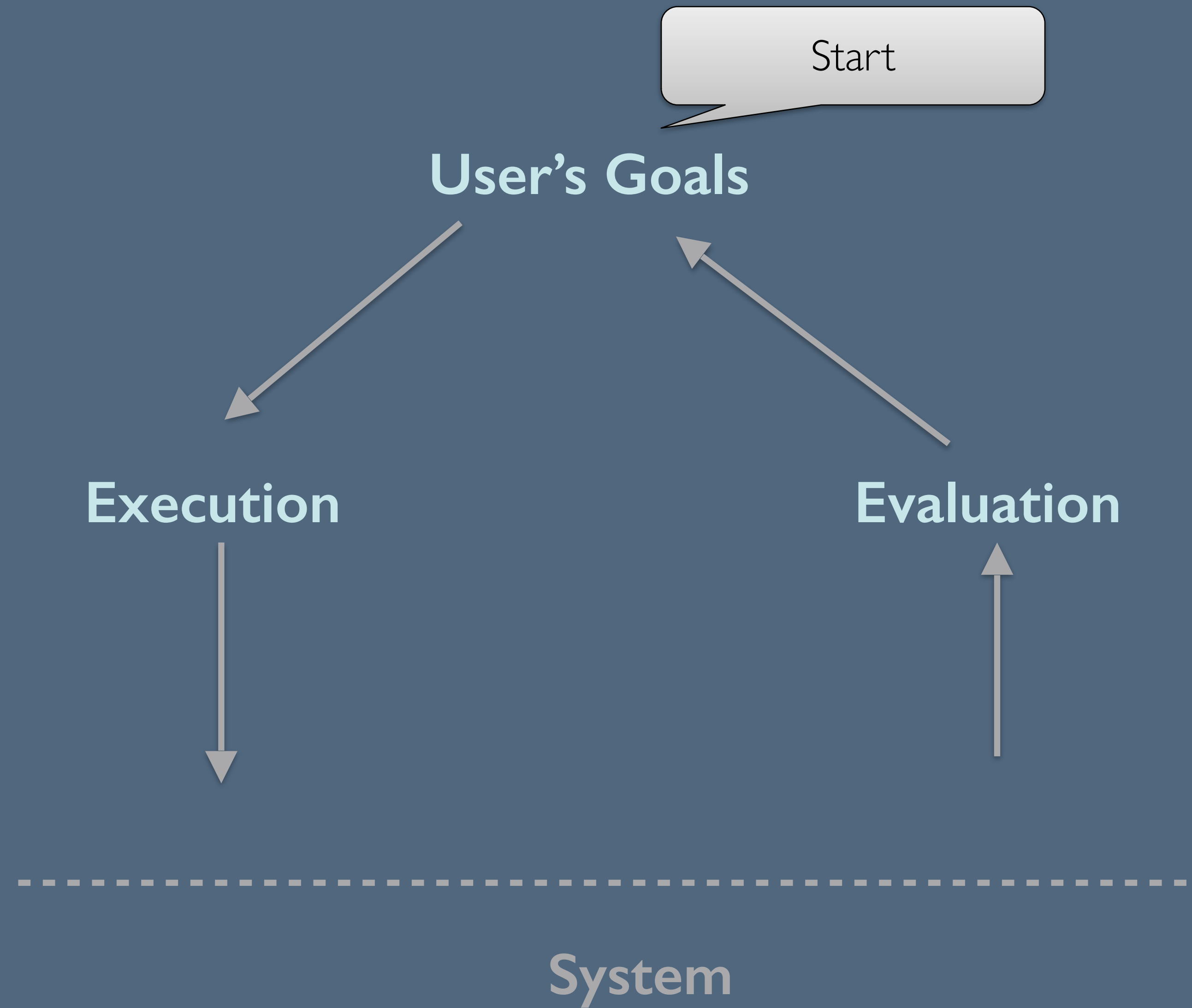
[Microsoft]

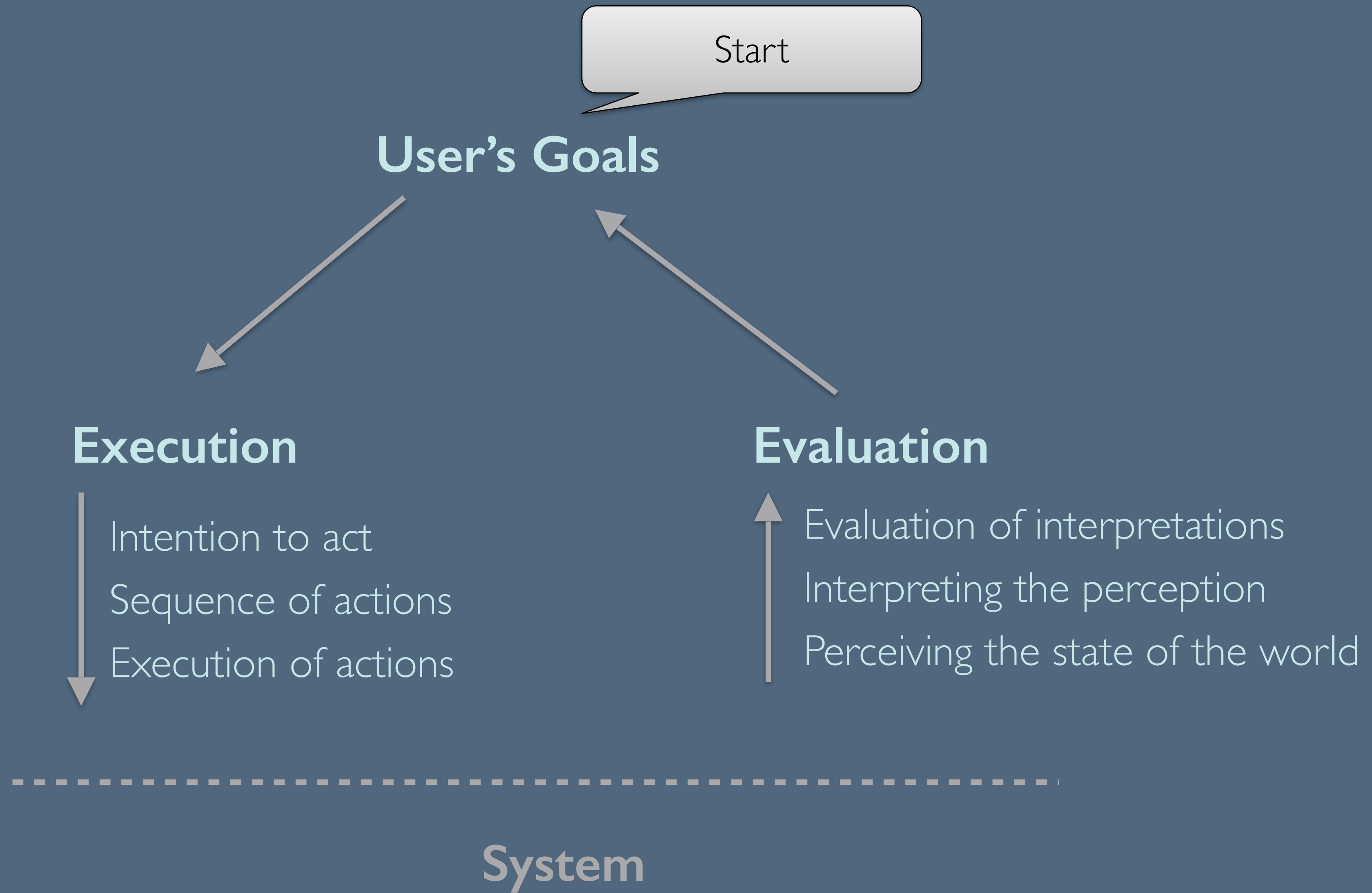
Gulfs between the person and the system

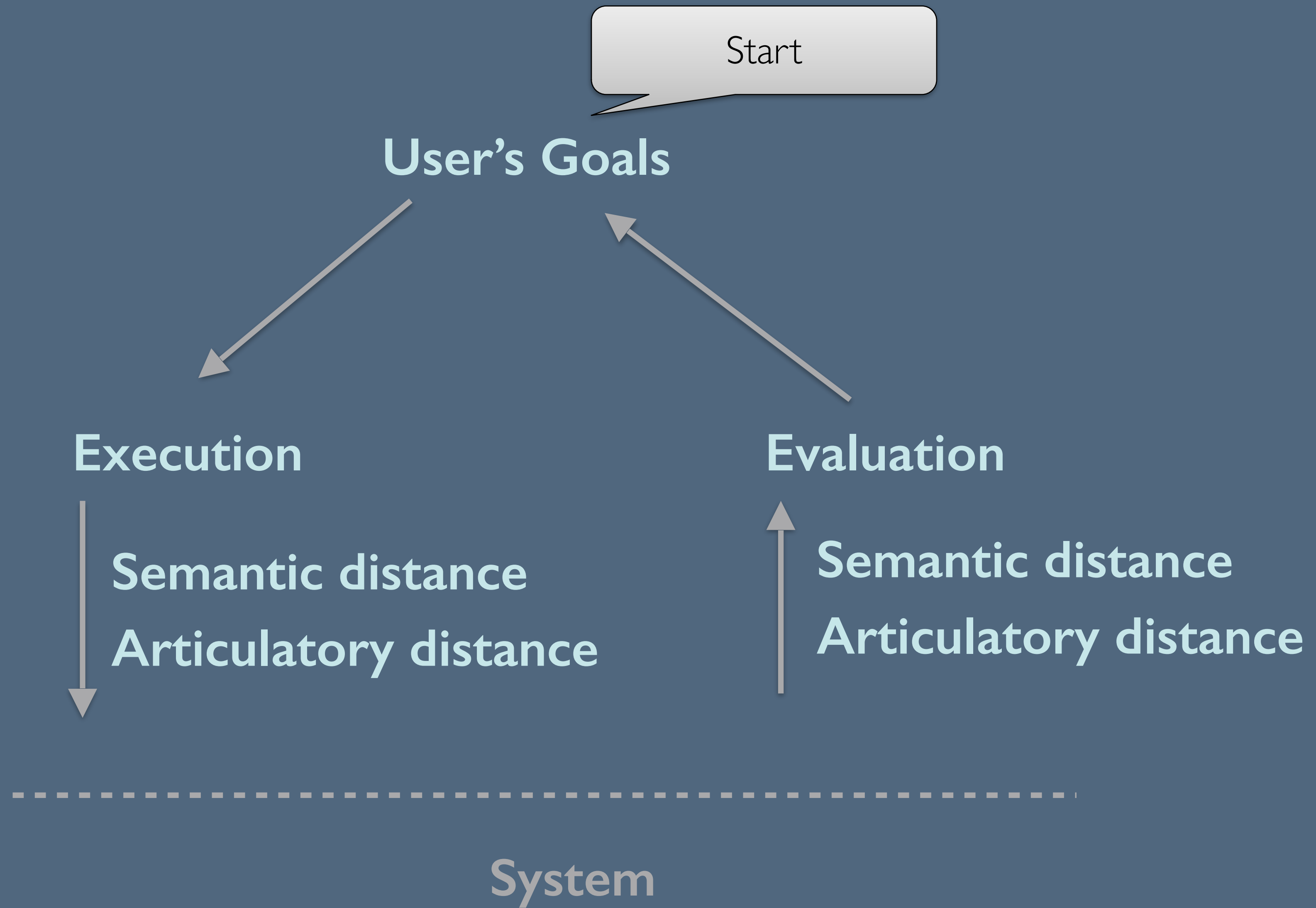
[Hutchins, Hollan and Norman 1985]

How many cognitive resources do I need to devote in order to translate from my goals to instructing the system and how do I interpret it's output?









Semantic and Articulatory Distances

Semantic distance is **cognitive** effort bridging the gap between the system's and user's meaning

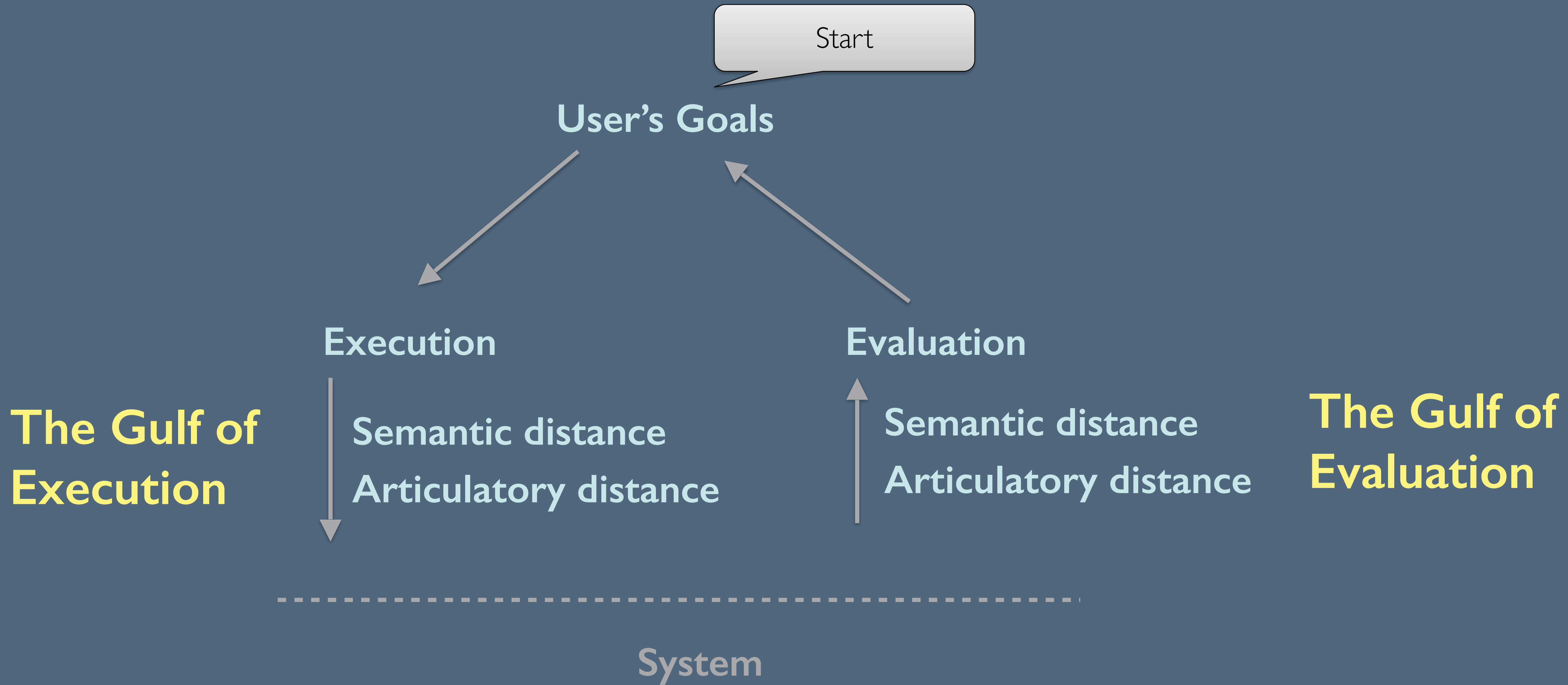
Is it possible to express what I'm trying to express?

What is this system trying to tell me?

Articulatory distance is **execution** effort translating between meaning and system input/output.

How much work is it for me to express my meaning using the system?

Is it straightforward for me to interpret the system's response symbols?



Always know which gap you're dealing with

If someone has to sit and figure out how to parse the symbols the system is presenting:

gulf of evaluation
(an articulatory distance)

If someone can decipher the symbols are, but can't figure out how the symbols relate to their goals:

also a gulf of evaluation
(but this time a semantic distance)

Always know which gap you're dealing with

If a nontechnical user knows which machine learning model they want to use but have to build it with raw tensors and have no idea how

gulf of execution (a semantic distance)

If that nontechnical expert has the right idea of what to command the system to do, but the function call is complicated (e.g. many parameters)

also a gulf of execution (but this time an articulatory distance)

Which gulf and distance is each design trying to address?

Google autocomplete

At-a-glance data summaries

Subway maps

DSLs (e.g., Python pandas

```
df.groupby('column').mean())
```

Gulf of
Evaluation

Gulf of
Execution

Semantic
distance

Articulatory
distance

Which gulf and distance is each design trying to address?

Google autocomplete

At-a-glance data summaries

Subway maps

DSLs (e.g., Python pandas

```
df.groupby('column').mean())
```

Gulf of Evaluation

Gulf of Execution

Semantic distance

Articulatory distance

Which gulf and distance is each design trying to address?

Google autocomplete

At-a-glance data summaries

Subway maps

DSLs (e.g., Python pandas

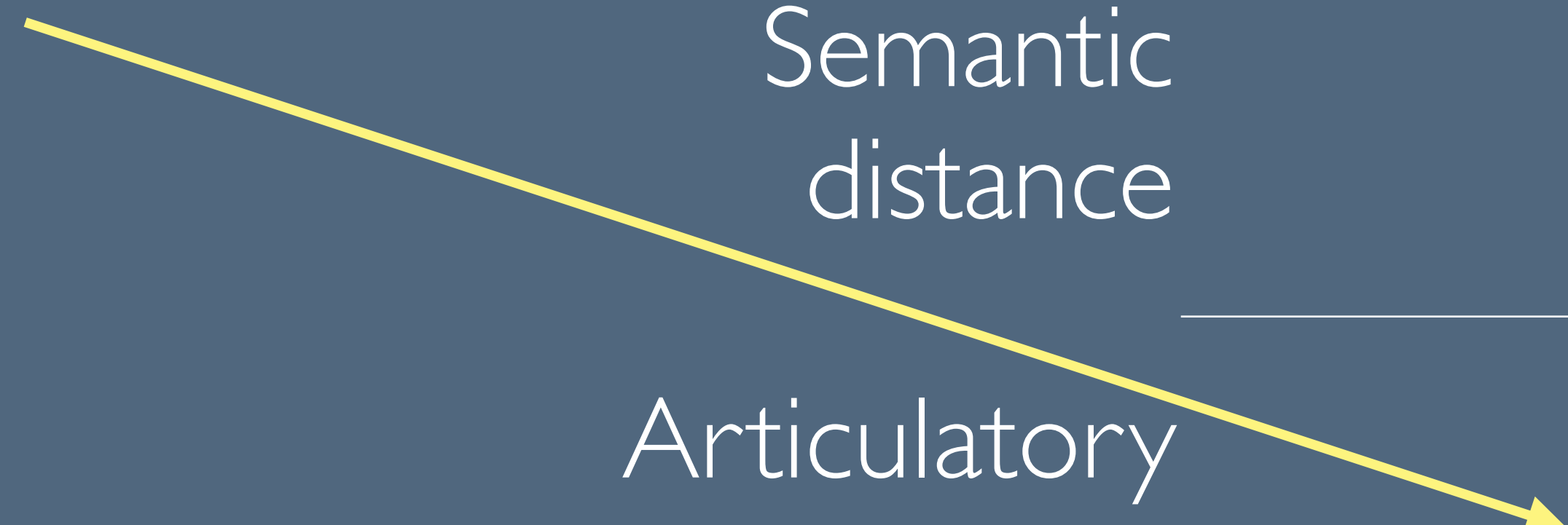
```
df.groupby('column').mean())
```

Gulf of
Evaluation

Gulf of
Execution

Semantic
distance

Articulatory
distance



Which gulf and distance is each design trying to address?

Google autocomplete

At-a-glance data summaries

Subway maps

DSLs (e.g., Python pandas

```
df.groupby('column').mean())
```

Gulf of Evaluation

Gulf of Execution

Semantic distance

Articulatory distance

Which gulf and distance is each design trying to address?

Google autocomplete

At-a-glance data summaries

Subway maps

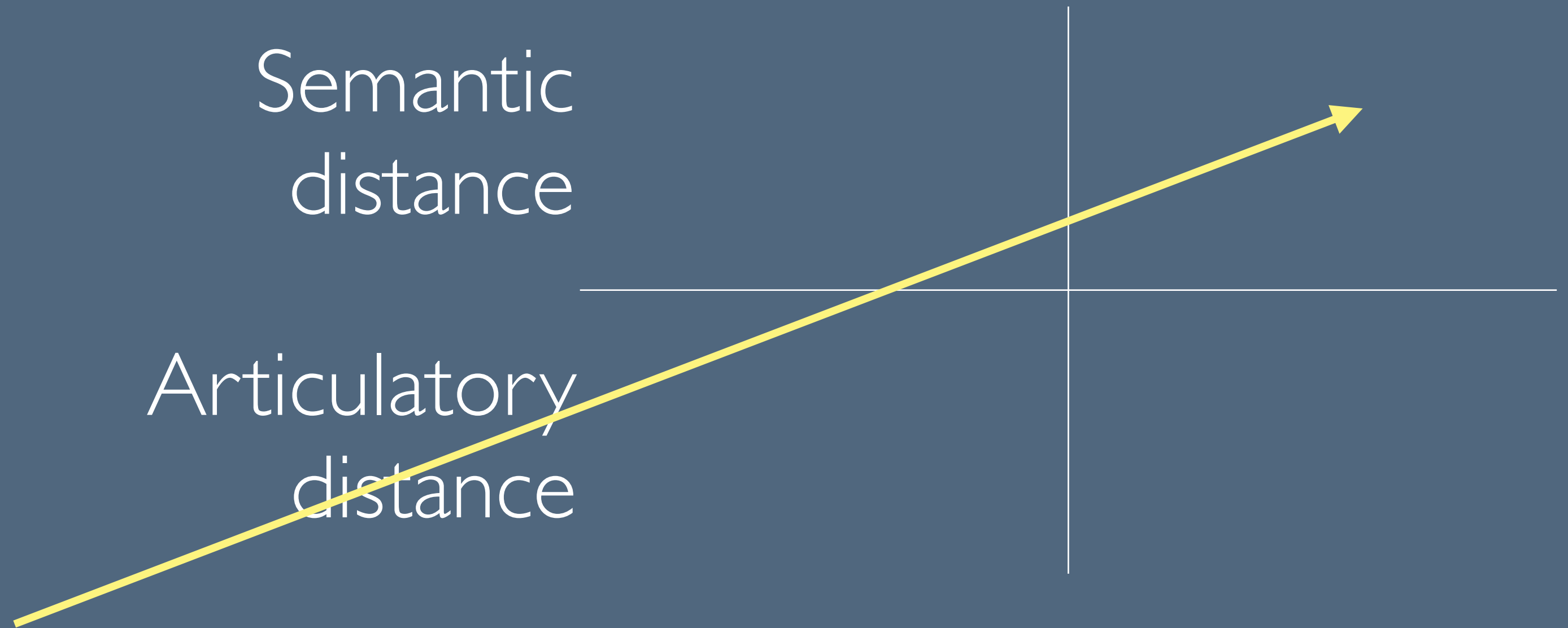
DSLs (e.g., Python pandas
`df.groupby('column').mean()`)

Gulf of Evaluation

Gulf of Execution

Semantic distance

Articulatory distance



Direct manipulation

[Hutchins, Hollan and Norman 1985]

Modern GUIs often adopt a metaphor of acting directly on the object of interest: **direct manipulation**. This reduces the gulfs.



Rather than scripts and code input, we act directly on the object
Rather than interpreting code output, the object itself has changed.

So, rather than aiming for “natural” interfaces, we should ask: **which gulf is this interface closing, and how?**

Gulfs in practice

1. **Gestural interaction:** the gulf of execution may remain wide, because either the semantic distance is large (Which gesture am I supposed to use again?), or the articulatory distance is large (It's hard to get the gesture recognized.)



2. **AI+HCI tools:** even if end-user tools reduce the gulf of execution, they may not reduce the gulf of evaluation (How do I interpret the AI errors?) or the next gulf of execution (How do I tweak the prev. result?)

Summary

These ideas help us fix the right problem rather than unreflectively following the design process.

Cognitive accounts can explain many challenges we face in design:

Design fixation: unnecessarily focusing on a subset of the design space

Analogical transfer: what do we see as related inspiration?

They can also help us be precise about how to improve design:

Gulfs of execution & evaluation: what needs to be reduced?

References

- Abtahi, Parastoo, et al. "Beyond Being Real: A Sensorimotor Control Perspective on Interactions in Virtual Reality." CHI Conference on Human Factors in Computing Systems. 2022.
- Buxton, Bill. Sketching user experiences: getting the design right and the right design. Morgan kaufmann, 2010.
- Choi, Hoon-Seok, and Leigh Thompson. "Old wine in a new bottle: Impact of membership change on group creativity." Organizational Behavior and human decision processes 98.2 (2005): 121-132.
- Dell, Nicola, et al. "'Yours is better!' participant response bias in HCI." Proceedings of the sigchi conference on human factors in computing systems. 2012.
- Duncker, Karl, and Lynne S. Lees. "On problem-solving." Psychological monographs 58.5 (1945): i.
- Gentner, Dedre, Jeffrey Loewenstein, and Leigh Thompson. "Learning and transfer: A general role for analogical encoding." Journal of educational psychology 95.2 (2003): 393
- Hollan, Jim, and Scott Stornetta. "Beyond being there." Proceedings of the SIGCHI conference on Human factors in computing systems. 1992.
- Hutchins, Edwin L., James D. Hollan, and Donald A. Norman. "Direct manipulation interfaces." Human-computer interaction 1.4 (1985): 311-338.
- Jansson, David G., and Steven M. Smith. "Design fixation." Design studies 12.1 (1991): 3-11.
- Kittur, Aniket, et al. "Scaling up analogical innovation with crowds and AI." Proceedings of the National Academy of Sciences 116.6 (2019): 1870-1877.

References

Luchins, Abraham S. "Mechanization in problem solving: The effect of Einstellung." *Psychological monographs* 54.6 (1942): i.

Maier, Norman R.F. "Reasoning in humans. II. The solution of a problem and its appearance in consciousness." *Journal of comparative Psychology* 12.2 (1931): 181.

Murray, Andrew. "Salvador Luria and Max Delbrück on random mutation and fluctuation tests." *Genetics* 202.2 (2016): 367-368.

Nicholas, Molly Jane, Sarah Serman, and Eric Paulos. "Creative and Motivational Strategies Used by Expert Creative Practitioners." *Creativity and Cognition*. 2022.

Norton, Michael I., Daniel Mochon, and Dan Ariely. "The IKEA effect: When labor leads to love." *Journal of consumer psychology* 22.3 (2012): 453-460.

Ross, Brian H. "Reminders and their effects in learning a cognitive skill." *Cognitive psychology* 16.3 (1984): 371-416.

Zirbel, Shannon A., et al. "Accommodating thickness in origami-based deployable arrays." *Journal of Mechanical Design* 135.11 (2013).