

Bridging the Gulfs of Execution and Evaluation

Discoverability

- **Users need to be able to discover**
 - **What a system can do**
 - **How to operate it**
- **This is how we bridge the Gulfs of Execution and Evaluation**
- **Execution: users can figure out what actions are possible**
- **Evaluation: users can discover whether actions were successful**

Supporting Discoverability

- **Affordances**
- **Signifiers**
- **Feedback**
- **Constraints**
- **Conceptual Models**

Affordances

- An “affordance” is a feature of an object or environment that indicates the possibility of action



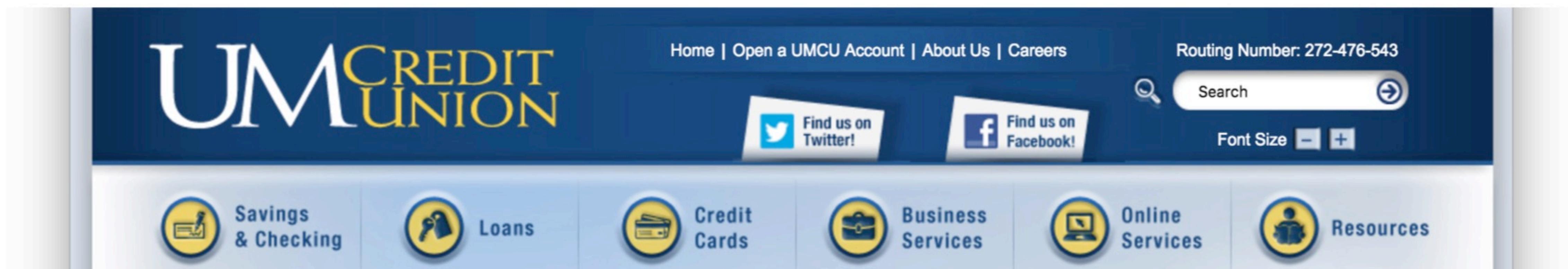
Affordances

- An “affordance” is a feature of an object or environment that indicates the possibility of action



Affordances

- An “affordance” is a feature of an object or environment that indicates the possibility of action



Signifiers

- An indication of *what action will occur and, in many cases, where the action can occur*
- A “sign”

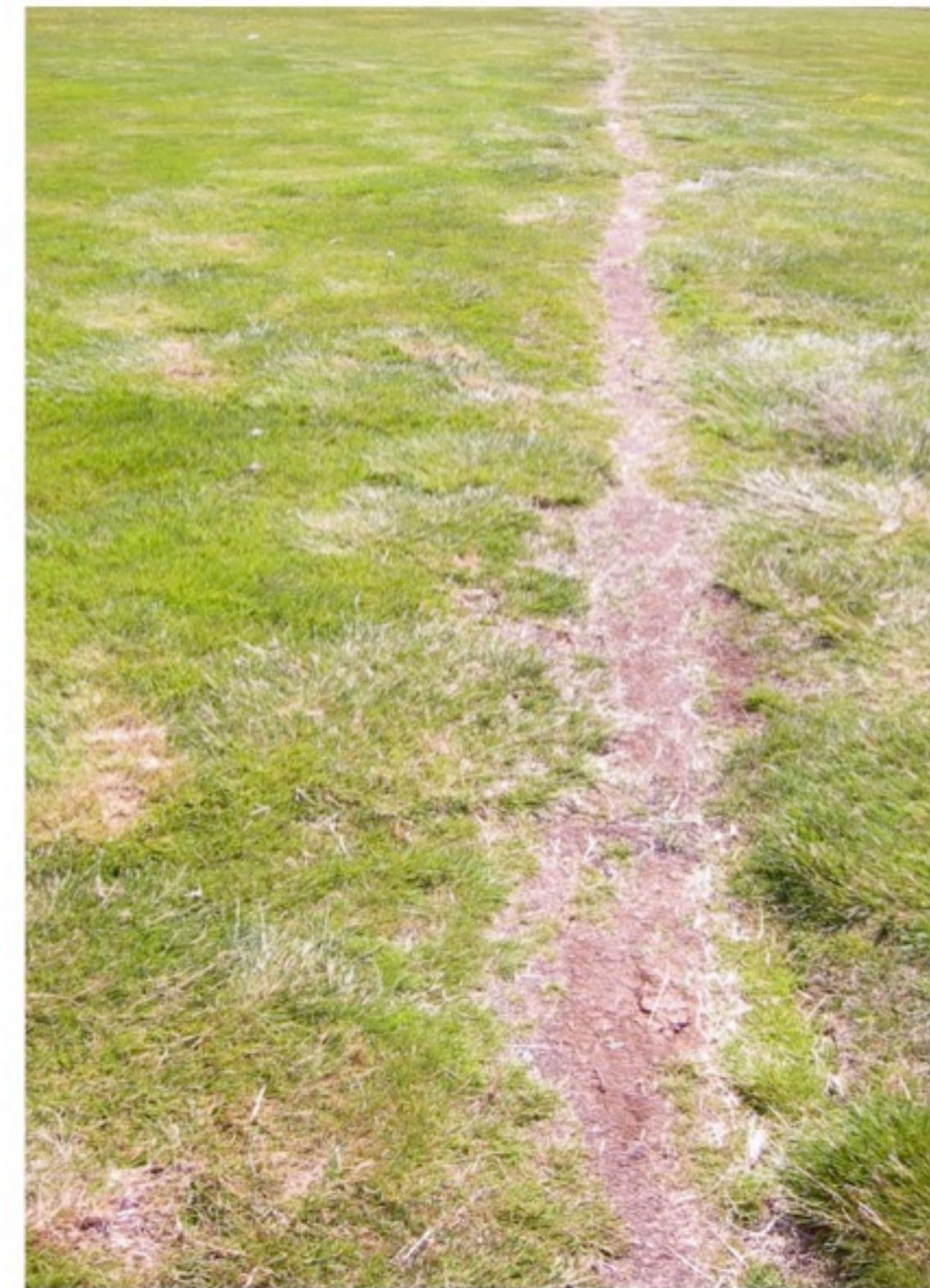
Signifiers

- An indication of *what action will occur and, in many cases, where the action can occur.*
- A “sign”



Signifiers

- An indication of *what action will occur and, in many cases, where the action can occur.*
- A “sign”

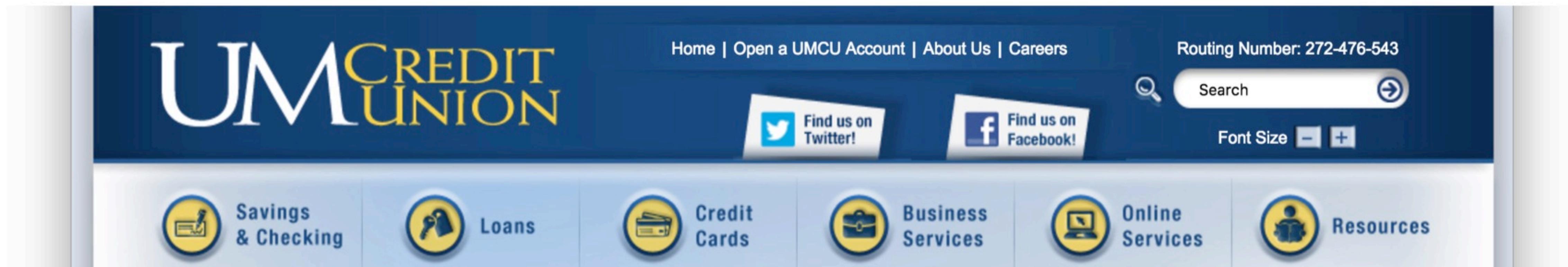


Affordances & Signifiers: Great Together!

- **Affordances, if done well, can communicate intuitively**
- **Signifiers are often necessary when many actions are possible**

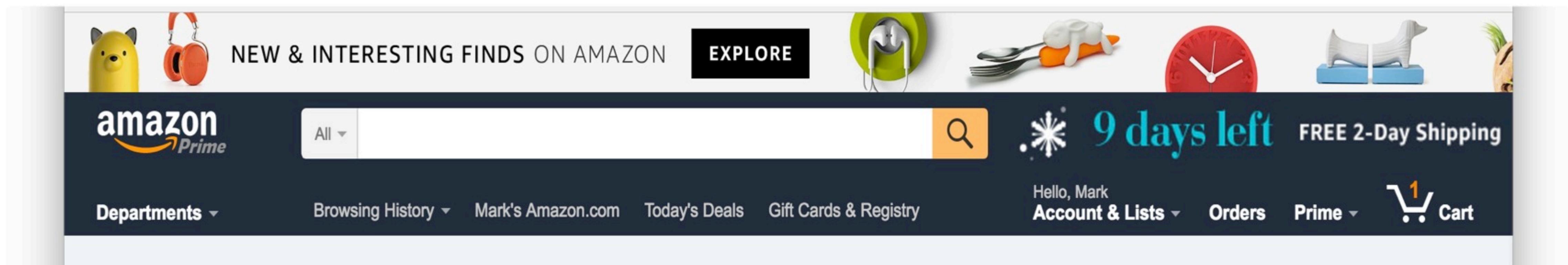
Affordances & Signifiers: Great Together!

- Affordances, if done well, can communicate intuitively
- Signifiers are often necessary when many actions are possible



Affordances & Signifiers: Great Together!

- Affordances, if done well, can communicate intuitively
- Signifiers are often necessary when many actions are possible
- Conventions and standards can reduce need for affordance



Affordances & Signifiers: Great Together!

- Affordances, if done well, can communicate intuitively
- Signifiers are often necessary when many actions are possible
- Conventions and standards can reduce need for affordance



Feedback

- **Users need to know that the system received their input**
- **Users need to know what the system did with their input**

Constraints

- **Unavailable actions should be disabled**
- **Limiting the total number of options will make selection easier**

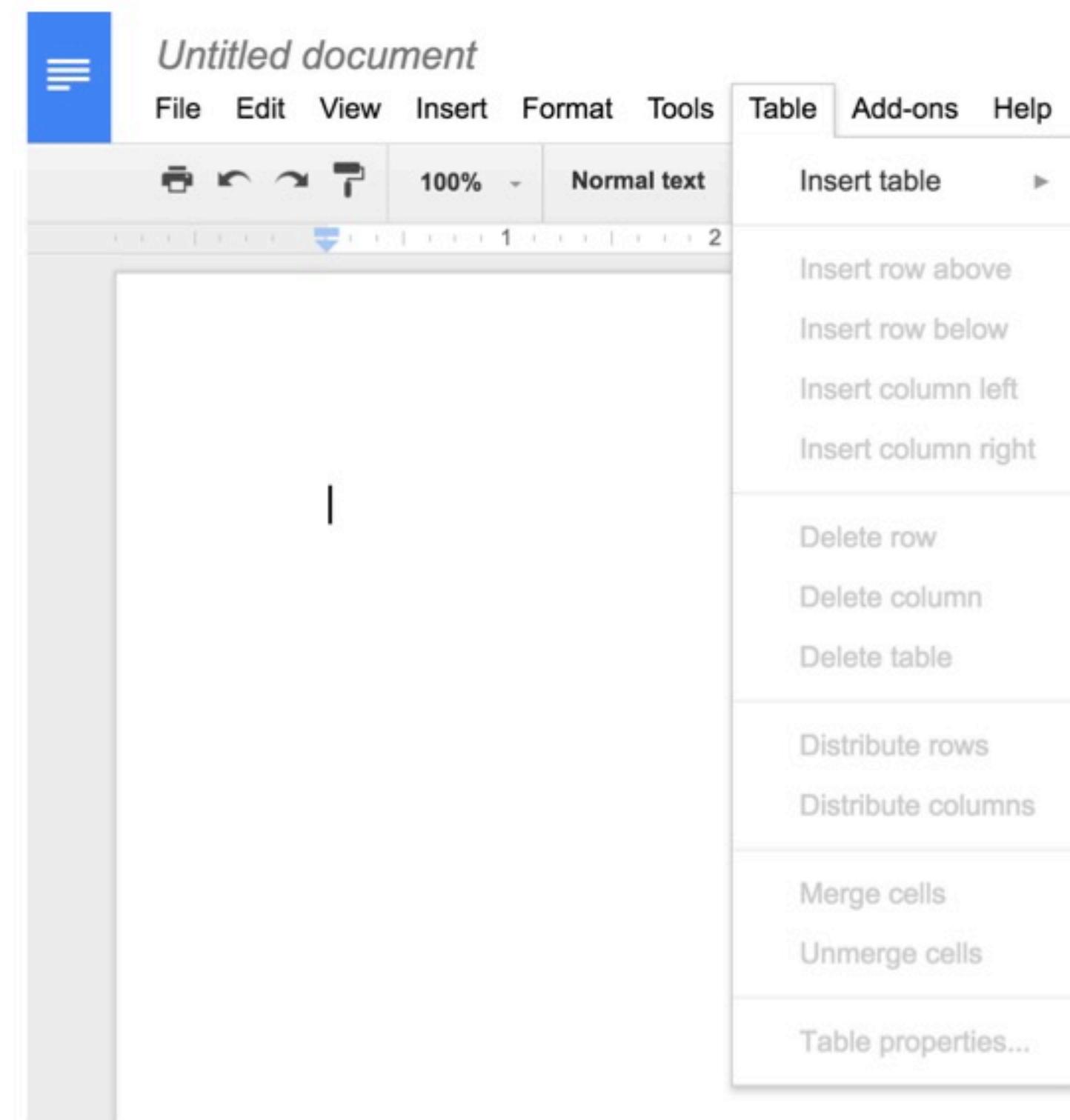
Constraints

- **Unavailable actions should be disabled**
- **Limiting the total number of options will make selection easier**



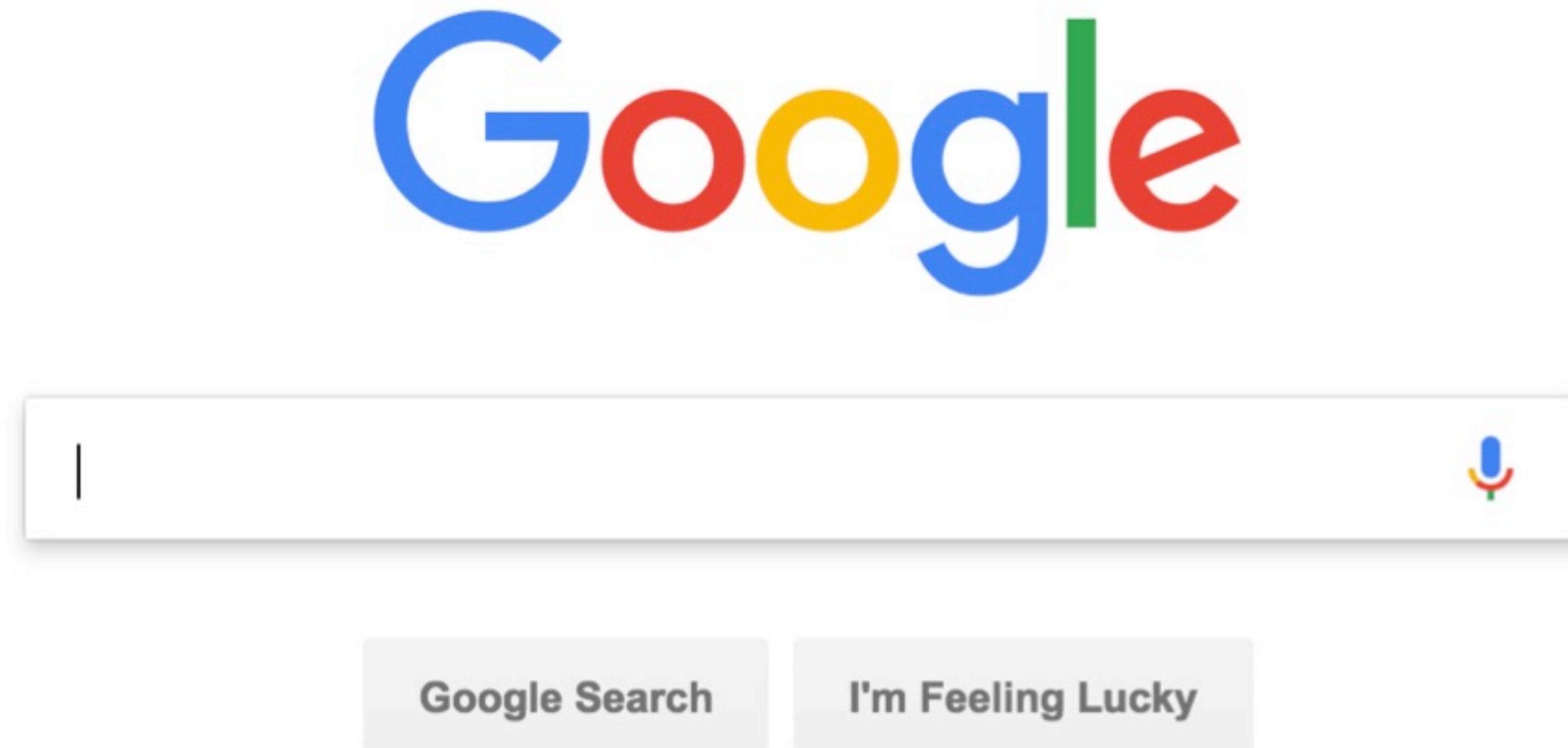
Constraints

- **Unavailable actions should be disabled**
- **Limiting the total number of options will make selection easier**



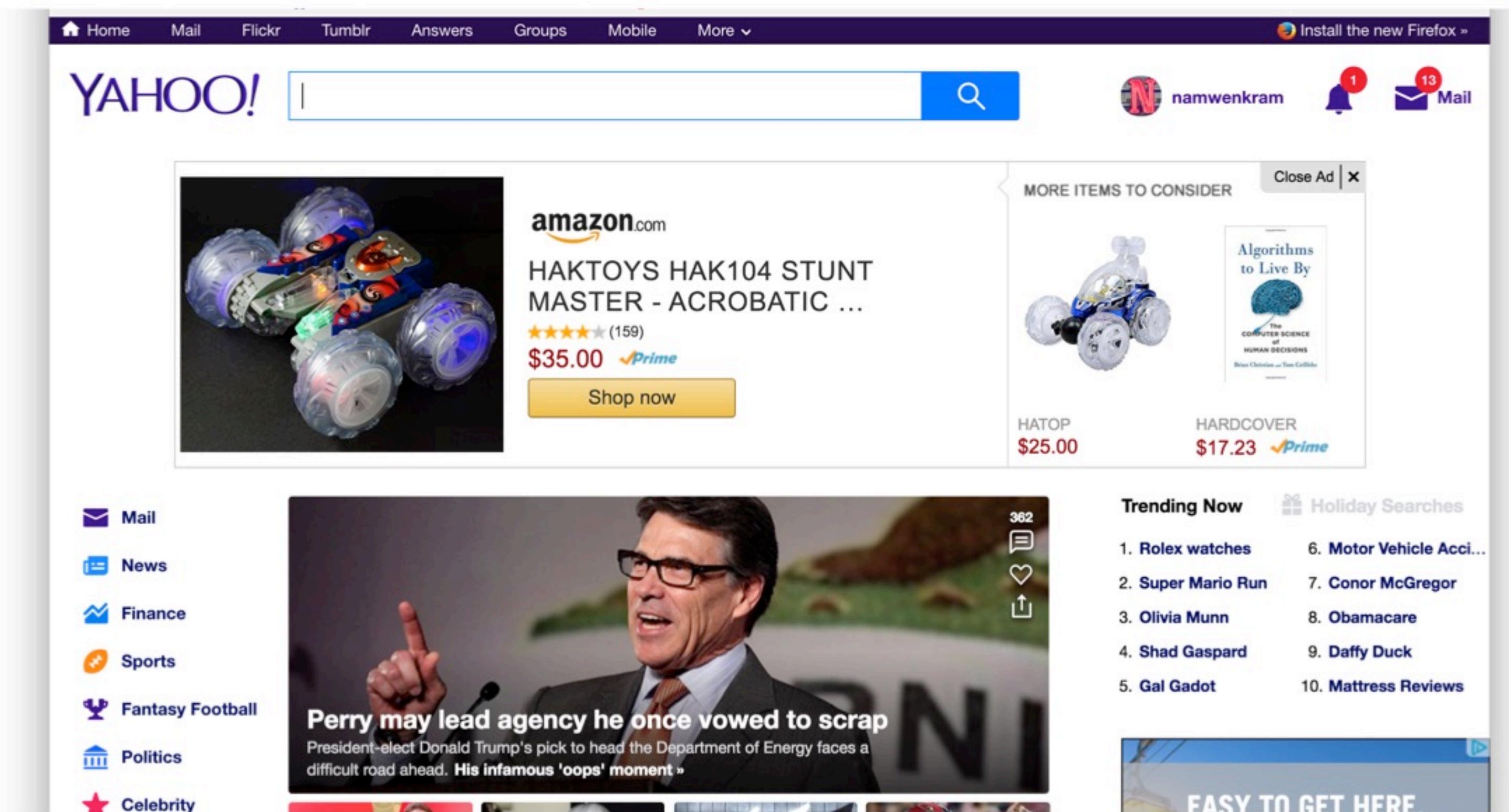
Constraints

- **Unavailable actions should be disabled**
- **Limiting the total number of options will make selection easier**



Constraints

- **Unavailable actions should be disabled**
- **Limiting the total number of options will make selection easier**



Conceptual Models

- Conceptual models support *simulation of future actions*
- Appropriate use of affordances, signifiers, feedback and constraints leads to *formation of accurate conceptual models*

Conceptual Models

- Conceptual models support *simulation of future actions*
- Appropriate use of affordances, signifiers, feedback and constraints leads to *formation of accurate conceptual models*

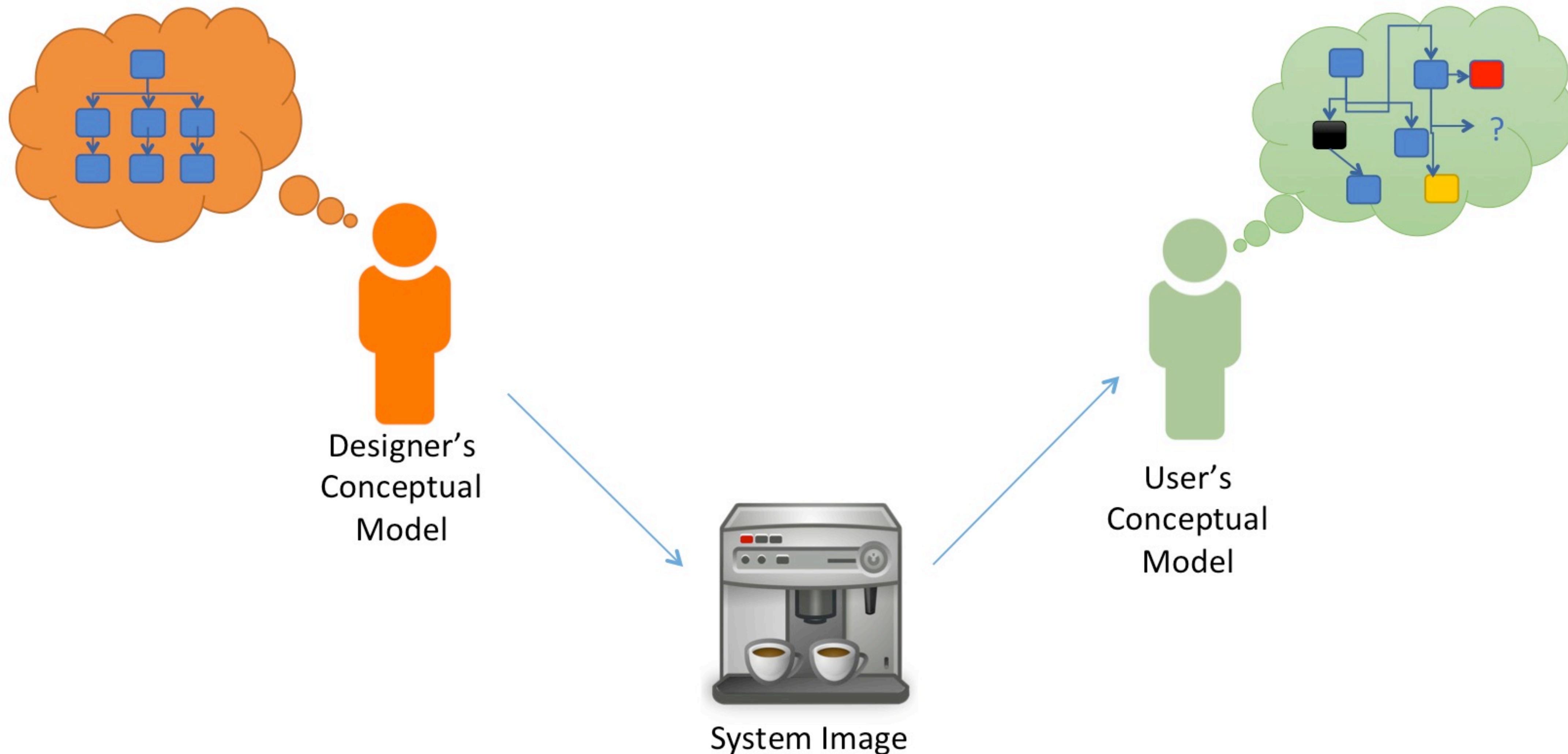


Conceptual Models

- Conceptual models support *simulation of future actions*
- Appropriate use of affordances, signifiers, feedback and constraints leads to *formation of accurate conceptual models*



Support Formation of Conceptual Models



Support Formation of Conceptual Models

- **Affordance**
- **Signifiers**
- **Feedback**
- **Constraints**

- **Consistency**
- **Metaphor**

Metaphors



Bridging the Gulfs of Execution and Evaluation

- **Discoverability**
 - What a system can do
 - How to do it
- **Conceptual Models**
 - What else the system is likely to be able to do
- **Affordances**
- **Signifiers**
- **Feedback**
- **Constraints**