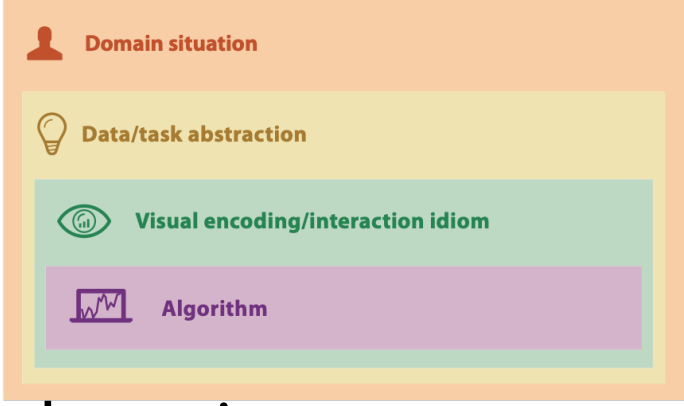
# Data Visualisation Tasks

Kamal Karlapalem

Spring 2024

Slides taken, reformatted, and used from Tamara Munzner (UBC, Canada)

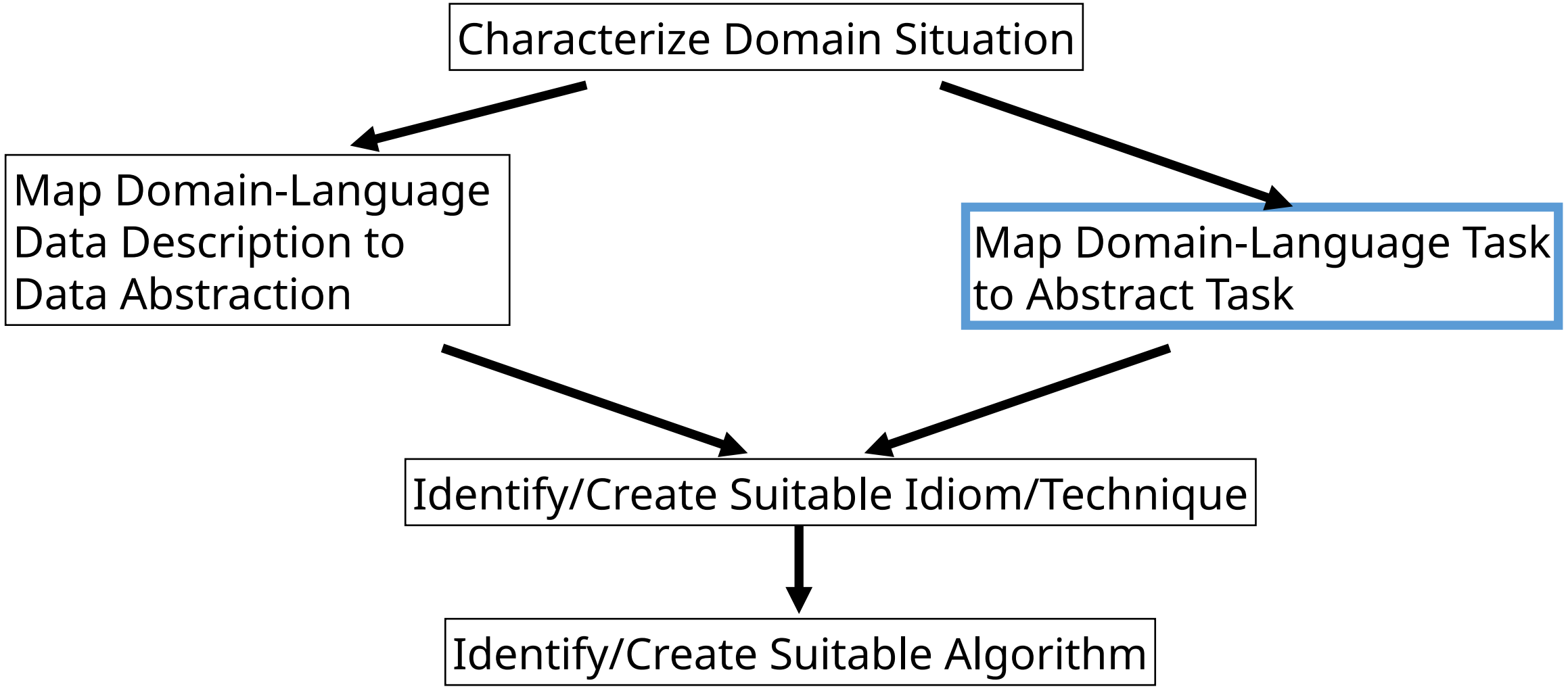
# From domain to abstraction




- Domain characterization: details of application domain
  - Group of users, target domain, their questions and data
    - Varies wildly by domain
    - Must be specific enough to get traction
  - Domain questions/problems
    - Break down into simpler abstract tasks
  - Abstraction: data and task
    - Map what and why into generalized terms
      - Identify tasks that users wish to perform or already do
      - Find data types that will support those tasks
        - Possibly transform/derive if need be

domain

# Design Process

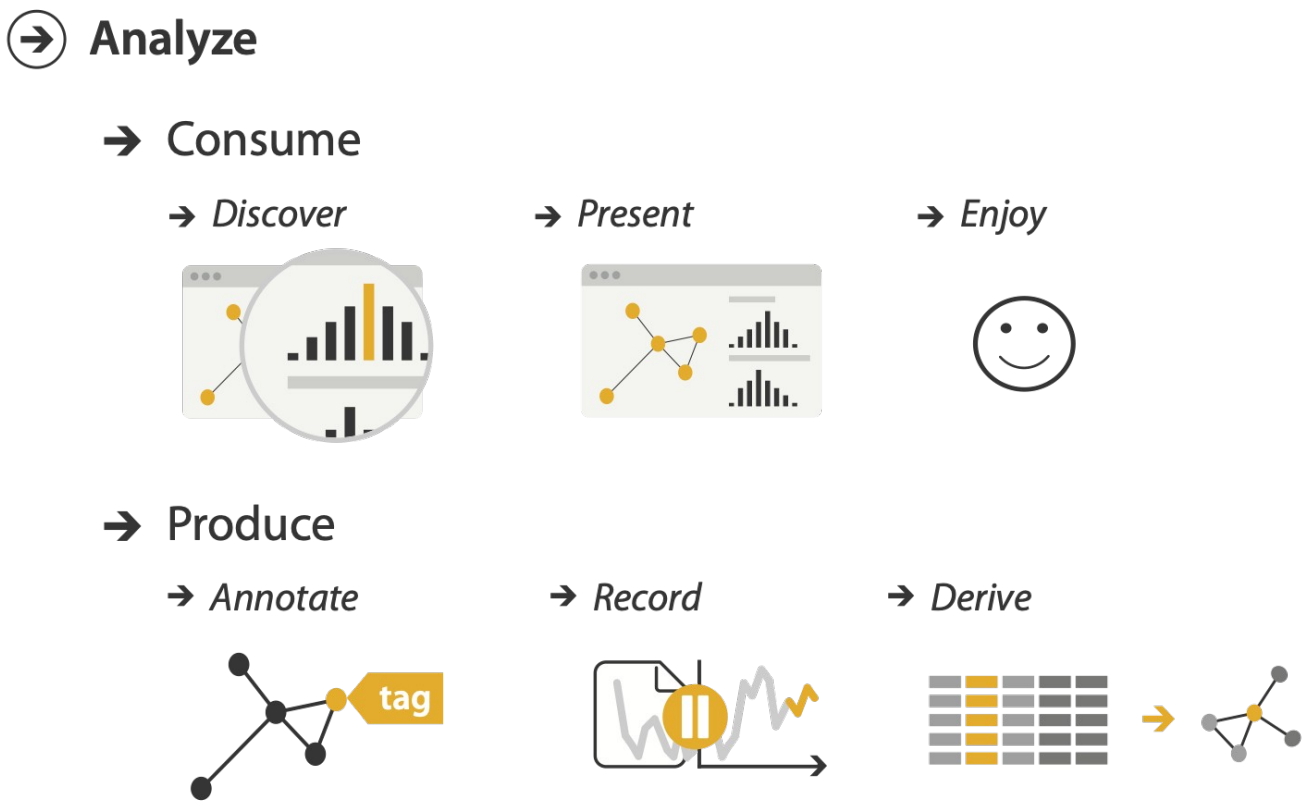


# Task abstraction : Actions and Targets

- Very high-level pattern
- Actions
  - Analyze
    - High-level choices
  - Search
    - Find a known/unknown item
  - Query
    - Find out about the characteristics of the item
- Targets
  - What is being acted on
- {action, target} pairs
  - Discover distribution
  - Compare trends
  - Locate outliers
  - Browse topology

# Actions: Analyze

- Consume
  - Discover vs present
    - Classic split
    - Explore vs explain
  - Enjoy
    - Newcomer
    - Casual, social
- Produce
  - Annotate, record
  - Derive
    - Crucial design choice

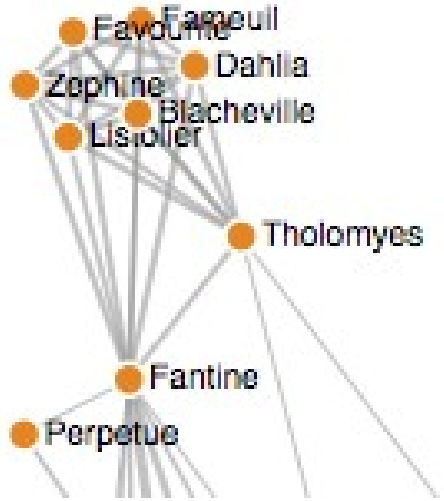


# Actions: Search

- What does the user know?
  - Target, location
- Lookup
  - Ex: word in a dictionary
    - Alphabetical order
- Locate
  - Ex: keys in your house
  - Ex: node in network
- Browse
  - Ex: books in the bookstore
- Explore
  - Ex: find cool neighborhood in a new city

**Search**

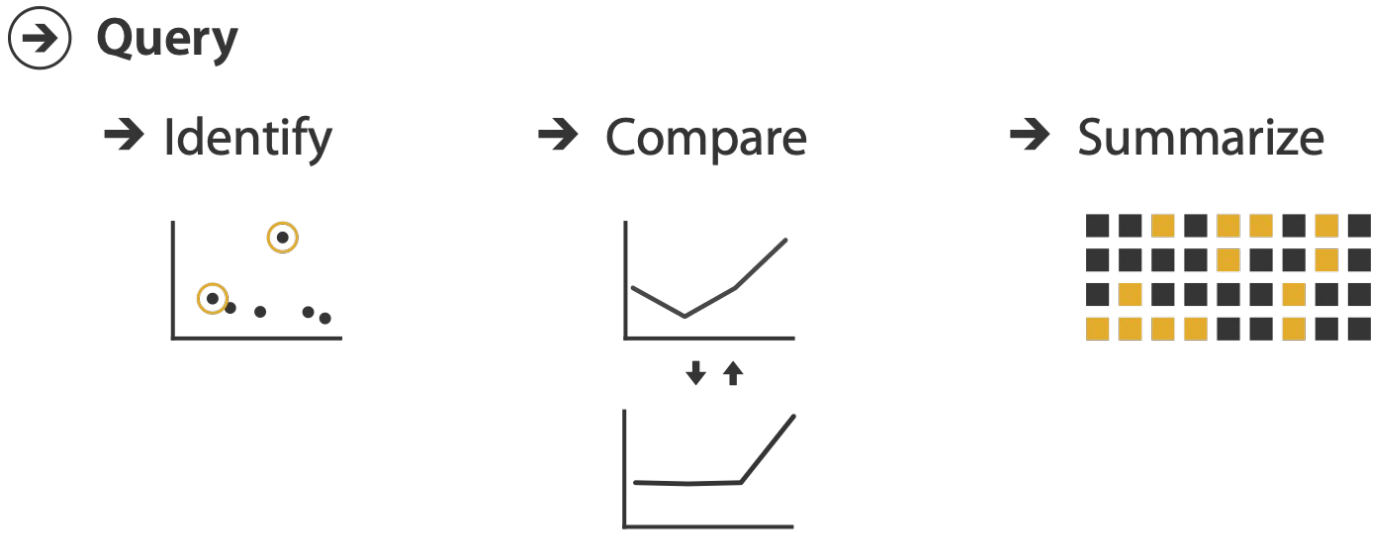
| | Target known | Target unknown |
|---|---|---|
| Location known | *Lookup* | *Browse* |
| Location unknown | *Locate* | *Explore* |




https://bl.ocks.org/heybignick/3faf257bbbbc7743bb72310d03b86ee8

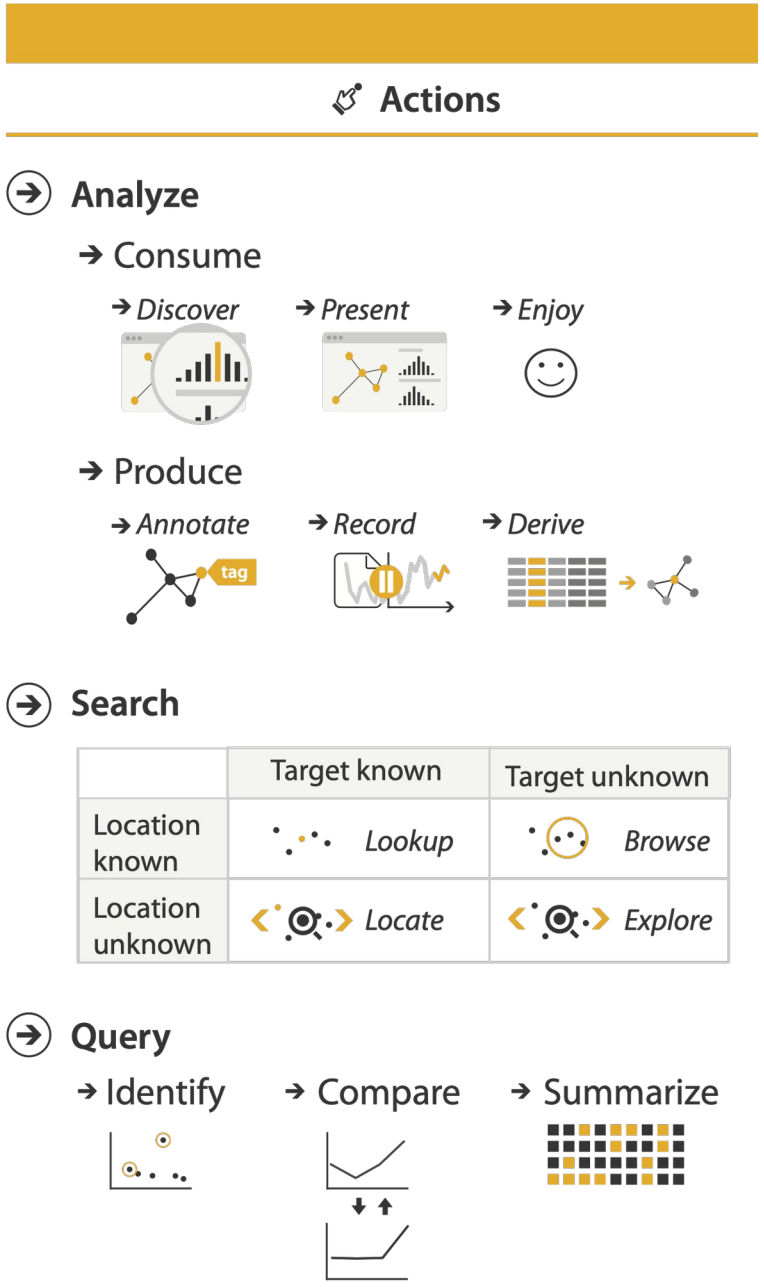
# Actions : Query

- How much of the data matters?
  - One : identify
  - Some : compare
  - All : summarize

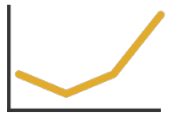
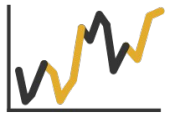
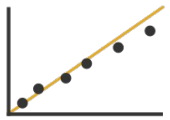
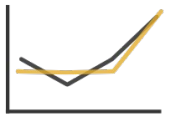
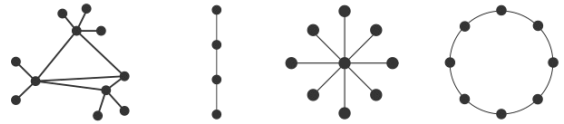
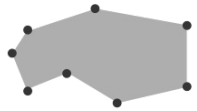


# Actions

- Independent choices for each of these three levels
  - Analyze, search, query
  - Mix and match



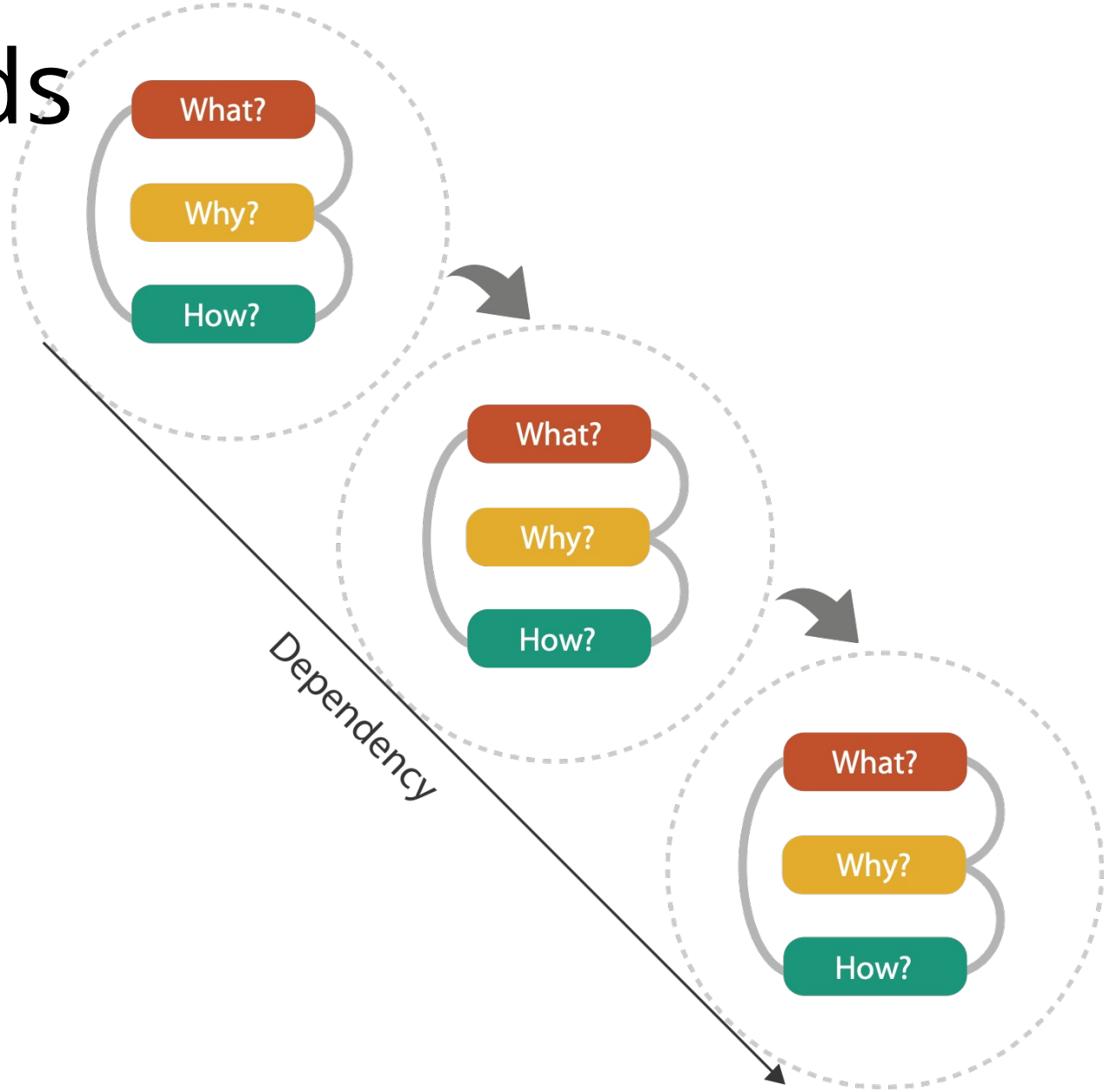
# Task abstractions : Targets

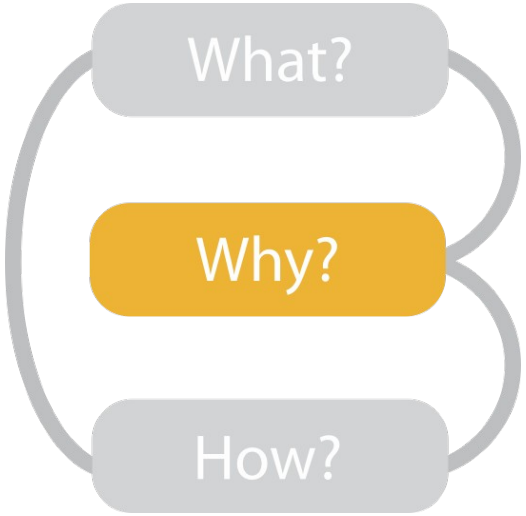
- **All Data**
  - Trends 
  - Outliers 
  - Features 
- **Attributes**
  - One
    - *Distribution* 
    - *Extremes* 
  - Many
    - *Dependency* 
    - *Correlation* 
    - *Similarity* 
- **Network Data**
  - Topology 
  - *Paths* 
- **Spatial Data**
  - Shape 

# Abstraction

- These {action, target} pairs are a good starting points for vocabulary
  - But sometime you will need more precision!
- Rule of thumb
  - Systematically remove all domain jargon
- Interplay : task and data abstraction
  - Need to use data abstraction within task abstraction
    - To specify your targets!
    - But task abstraction can lead you to transform the data
  - Iterate back and forth
    - First pass data, first pass task, second pass data, …

# Means and ends






- {action, target} pairs
  - *discover distribution*
  - *compare trends*
  - *locate outliers*
  - *browse topology*

## Why?

### Actions

**Analyze**

- Consume
  - *Discover*
  - *Present*
  - *Enjoy*
- Produce
  - *Annotate*
  - *Record*
  - *Derive*

**Search**

| | Target known | Target unknown |
|---|---|---|
| Location known | *Lookup* | *Browse* |
| Location unknown | *Locate* | *Explore* |

**Query**

- Identify
- Compare
- Summarize

### Targets

**All Data**

- Trends
- Outliers
- Features

**Attributes**

- One
  - *Distribution*
  - *Extremes*
- Many
  - *Dependency*
  - *Correlation*
  - *Similarity*

**Network Data**

- Topology
- *Paths*

**Spatial Data**

- Shape