

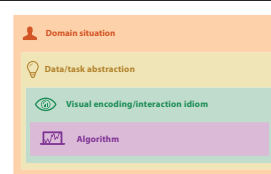
Visualization Analysis & Design

Task Abstraction (Ch 3)

Tamara Munzner
 Department of Computer Science
 University of British Columbia
 @tamaramunzner



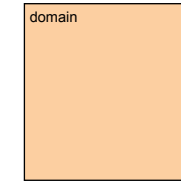
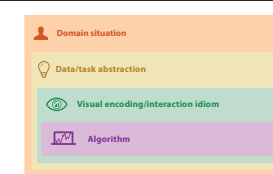
From domain to abstraction



2

From domain to abstraction

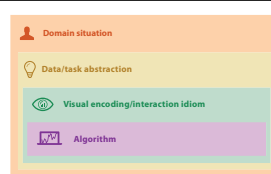
- domain characterization: details of application domain



3

From domain to abstraction

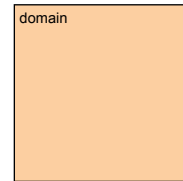
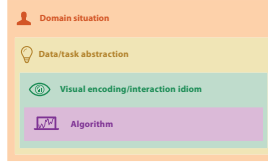
- domain characterization: details of application domain
 - group of users, target domain, their questions & data
 - varies wildly by domain
 - must be specific enough to get traction



4

From domain to abstraction

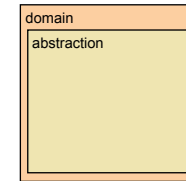
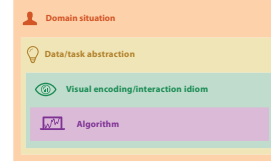
- domain characterization: details of application domain
 - group of users, target domain, their questions & data
 - varies wildly by domain
 - must be specific enough to get traction
 - domain questions/problems
 - break down into simpler abstract tasks



5

From domain to abstraction

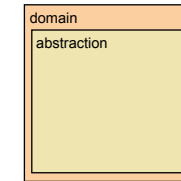
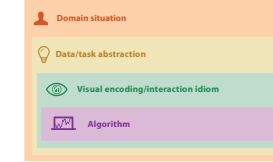
- domain characterization: details of application domain
 - group of users, target domain, their questions & data
 - varies wildly by domain
 - must be specific enough to get traction
 - domain questions/problems
 - break down into simpler abstract tasks
- abstraction: data & task
 - map *what* and *why* into generalized terms



6

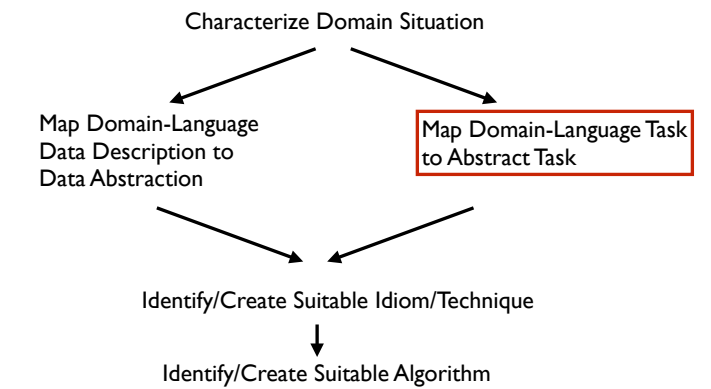
From domain to abstraction

- domain characterization: details of application domain
 - group of users, target domain, their questions & data
 - varies wildly by domain
 - must be specific enough to get traction
 - domain questions/problems
 - break down into simpler abstract tasks
- abstraction: data & 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



7

Design process



8

Task abstraction: Actions and targets

- very high-level pattern
- {action, target} pairs
 - discover distribution
 - compare trends
 - locate outliers
 - browse topology

9

Task abstraction: Actions and targets

- very high-level pattern
- {action, target} pairs
 - discover distribution
 - compare trends
 - locate outliers
 - browse topology
- actions
 - analyze
 - high-level choices
 - search
 - find a known/unknown item
 - query
 - find out about characteristics of item

10

Task abstraction: Actions and targets

- very high-level pattern
- {action, target} pairs
 - discover distribution
 - compare trends
 - locate outliers
 - browse topology
- actions
 - analyze
 - high-level choices
 - search
 - find a known/unknown item
 - query
 - find out about characteristics of item
- targets
 - what is being acted on

11

Actions: Analyze

- consume
 - discover vs present
 - classic split
 - aka explore vs explain
 - enjoy
 - newcomer
 - aka casual, social
- produce
 - annotate, record
 - derive
 - crucial design choice

12

Actions: Search

Actions: Search

- what does user know?
 - target, location

	Target known	Target unknown
Location known	Lookup	Browse
Location unknown	Locate	Explore

14

Actions: Search

- what does user know?
 - target, location
- lookup
 - ex: word in dictionary
 - alphabetical order

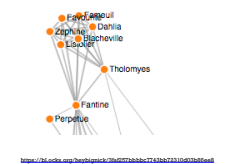
	Target known	Target unknown
Location known	Lookup	Browse
Location unknown	Locate	Explore

15

Actions: Search

- what does user know?
 - target, location
- lookup
 - ex: word in dictionary
 - alphabetical order
- locate
 - ex: keys in your house
 - ex: node in network

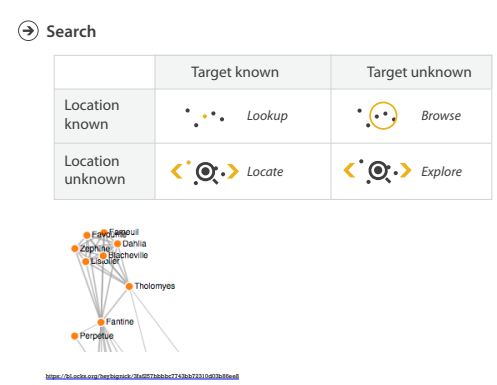
	Target known	Target unknown
Location known	Lookup	Browse
Location unknown	Locate	Explore



16

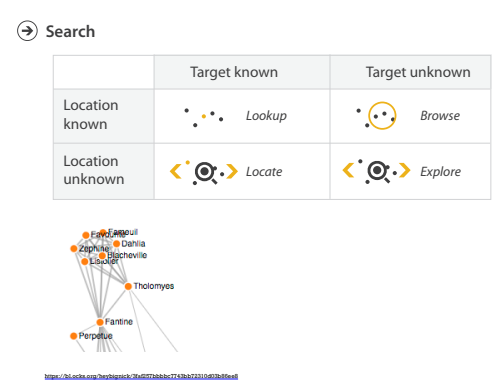
Actions: Search

- what does user know?
 - target, location
- lookup
 - ex: word in dictionary
 - alphabetical order
- locate
 - ex: keys in your house
 - ex: node in network
- browse
 - ex: books in bookstore



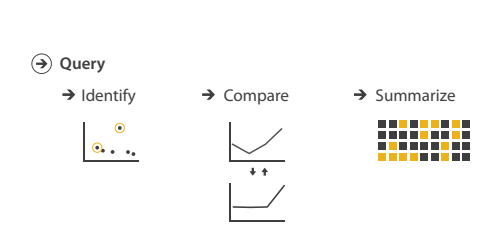
Actions: Search

- what does user know?
 - target, location
- lookup
 - ex: word in dictionary
 - alphabetical order
- locate
 - ex: keys in your house
 - ex: node in network
- browse
 - ex: books in bookstore
- explore
 - ex: find cool neighborhood in new city



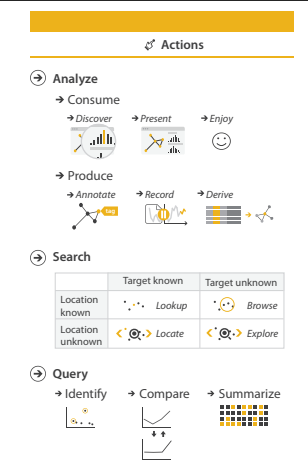
Actions: Query

- 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

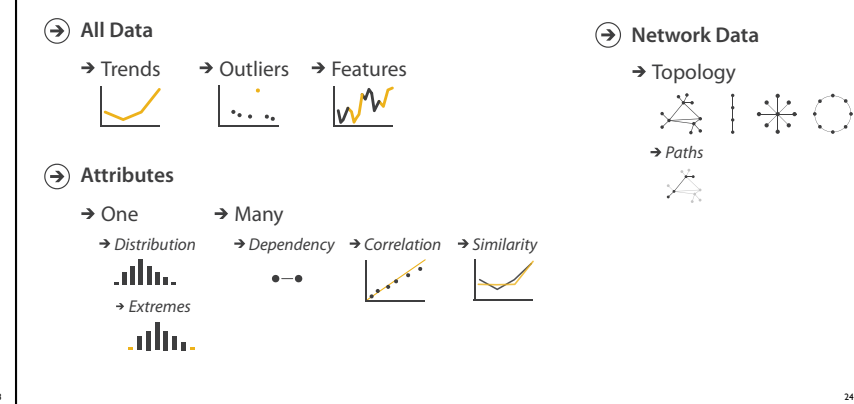
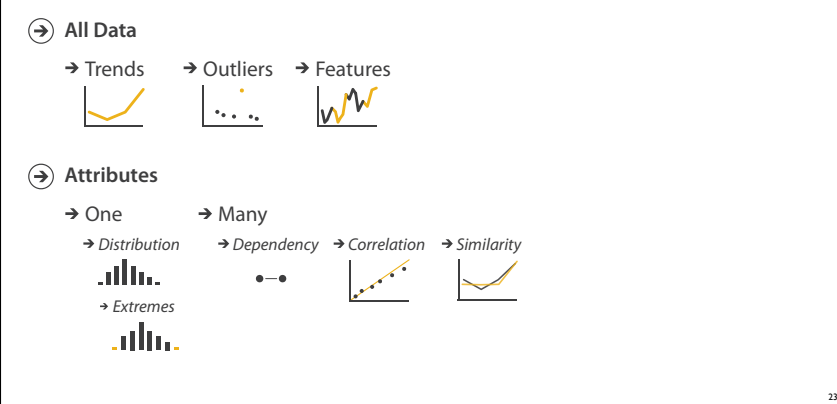
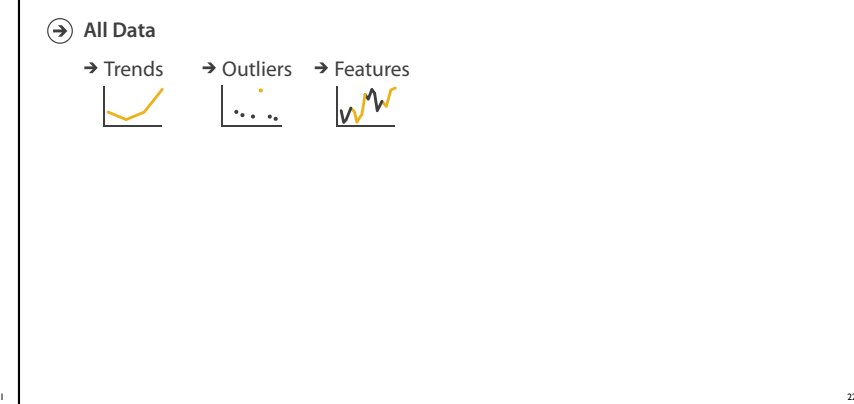


Task abstraction: Targets

Task abstraction: Targets

Task abstraction: Targets

Task abstraction: Targets

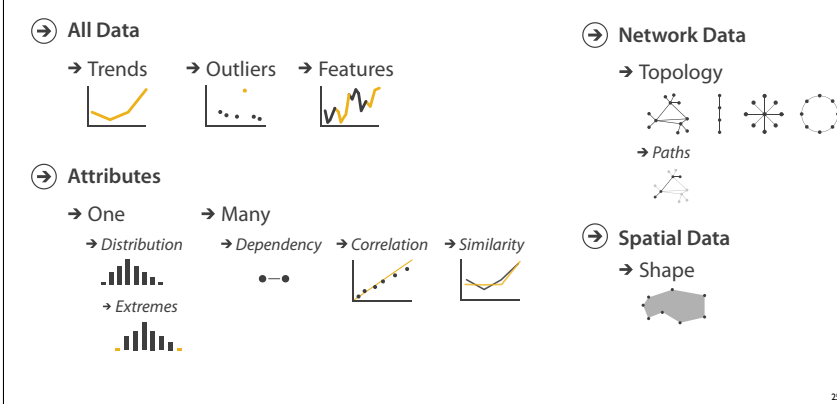


Task abstraction: Targets

Abstraction

Means and ends

Why?



- these {action, target} pairs are good starting point for vocabulary
 - but sometimes you'll 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, ...

