

WHY: TASK ABSTRACTION

Bùi Tiến Lên

01/01/2020



KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Contents



1. **Actions**

2. **Targets**

3. **How**

4. **Analyzing and Deriving: Examples**



The Big Picture

Actions

Analyze
Produce
Search
Query

Targets

How

Analyzing and Deriving: Examples

Comparing Two Idioms
Deriving One Attribute
Deriving Many New Attributes

Why?

Actions

Targets

→ Analyze

→ Consume



→ Produce



→ Search

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

→ Query

→ Identify



→ Compare



→ Summarize



→ All Data

→ Trends



→ Outliers



→ Features



→ Attributes

→ One

→ Distribution



→ Extremes



→ Many

→ Dependency



→ Correlation

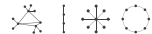


→ Similarity



→ Network Data

→ Topology



→ Paths



→ Spatial Data

→ Shape



What?

Why?

How?

Why Analyze Tasks Abstractly?



- This framework encourages you to consider tasks in abstract form, rather than the domain-specific way that users typically think about them.
- Transforming task descriptions from domain-specific language into abstract form allows you to reason about similarities and differences between them.

Who: Designer or User



- It's sometimes useful to augment an analysis instance specification by indicating who has a goal or makes a design choice: the designer of the vis or the end user of the vis.



Actions

- Analyze
- Produce
- Search
- Query

Actions



Three levels of **actions** that define user goals

- The high-level choices describe how the vis is being used to **analyze**, either to consume existing data or to also produce additional data.
- The mid-level choices cover what kind of **search** is involved, in terms of whether the target and location are known or not.
- The low-level choices pertain to the kind of **query**: does the user need to identify one target, compare some targets, or summarize all of the targets?

Analyze



- At the highest level, the framework distinguishes between two possible goals of people who want to analyze data using a vis tool: users might want only to consume existing information or also to actively produce new information.
- The most common use case for vis is for the user to **consume** information that has already been generated as data stored in a format amenable to computation

Discover



- The **discover** goal refers to using vis to find new knowledge that was not previously known.
- This usage includes the goal of finding completely new things; that is, the outcome is to **generate** a new hypothesis.
- It also includes the goal of figuring out whether a conjecture is true or false; that is, to **verify**—or disconfirm—an existing hypothesis.

Present

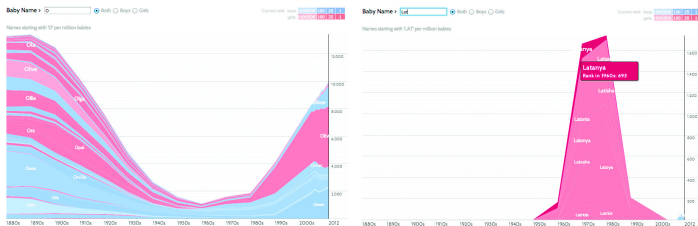


- The **present** goal refers to the use of vis for the succinct communication of information, for telling a story with data, or guiding an audience through a series of cognitive operations.
- Presentation using vis may take place within the context of decision making, planning, forecasting, and instructional processes.

Enjoy



- The **enjoy** goal refers to casual encounters with vis. In these contexts, the user is not driven by a previously pressing need to verify or generate a hypothesis but by curiosity that might be both stimulated and satisfied by the vis
- Name Voyager, a vis tool originally intended for parents focused deciding on what to name their expected baby, ended up being used by many nonparents to analyze historical trends for their own enjoyment. Left: Names starting with 'O' had a notable dip in popularity in the middle of the century. Right: Names starting with 'LAT' show a trend of the 1970s



Produce



- In contrast to using vis only for consuming existing information, in the **produce** case the intent of the user is to generate new material.
- Often the goal with produce is to produce output that is used immediately, as input to a next instance.
- Sometimes the user intends to use this new material for some other vis-related task later on, such as discovery or presentation.

Annotate



- The **annotate** goal refers to the addition of graphical or textual annotations associated with one or more preexisting visualization elements, typically as a manual action by the user.
- When an annotation is associated with data items, the annotation could be thought of as a new attribute for them.

Record



- The **record** goal saves or captures visualization elements as persistent artifacts.
- These artifacts include screen shots, lists of bookmarked elements or locations, parameter settings, interaction logs, or annotations.
- Graphical history recorded during an analysis session with Tableau.

The screenshot shows the 'Worksheet History' panel in Tableau. It includes a 'Filters' section, a 'Graphic Type' dropdown set to 'All', and a 'Bookmarks' section with a 'Show Only' checkbox. Below these are 'Export' and 'Reset' buttons. The main area displays a list of actions with their corresponding visualizations:

	Major Ma..	Small Mar..
Central	1,683,579	1,563,045
East	1,628,963	624,021
South	279,067	760,398
West	1,197,854	2,617,430

The actions and their visualizations are:

- Add Inventory (Table)
- Show Me! (Bar chart)
- Move Market Size to Columns (Bar chart)
- Add Product to Columns (Line chart)

Derive



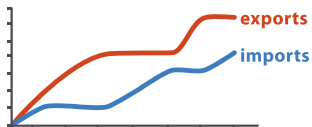
- The **derive** goal is to produce new data elements based on existing data elements.
- New attributes can be derived from information contained within existing ones, or data can be transformed from one type into another.
- Deriving new data is a critical part of the vis design process.



Example

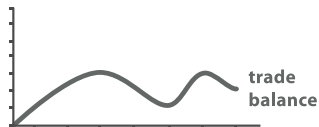
Derived attributes can be directly visually encoded.

- (a) Two original data attributes are plotted, imports and exports.
- (b) The quantitative derived attribute of trade balance, the difference between the two originals, can be plotted directly.



Original Data

(a)



$$\text{trade balance} = \text{exports} - \text{imports}$$

Derived Data

(b)

Lookup



- If users already know both what they're looking for and where it is, then the search type is simply **lookup**

Actions

Analyze

Produce

Search

Query

Targets

How

Analyzing and Deriving: Examples


Comparing Two Idioms

Deriving One Attribute

Deriving Many New Attributes

Locate

- To find a known target at an unknown location, the search type is **locate**: that is, find out where the specific object is



18

Browse



- In contrast, the exact identity of a search target might not be known in advance; rather, it might be specified based on characteristics. In this case, users are searching for one or more items that fit some kind of specification, such as matching up with a particular range of attribute values
- When users don't know exactly what they're looking for, but they do have a location in mind of where to look for it, the search type is **browse**.

Explore



- When users are not even sure of the location, the search type is **explore**.
- It entails searching for characteristics without regard to their location, often beginning from an overview of everything

Query



- Once a target or set of targets for a search has been found, a low-level user goal is to query these targets at one of three scopes: **identify**, **compare**, or **summarize**
- The progression of these three corresponds to an increase in the amount of search targets under consideration: one, some, or all

Identify



- The scope of **identify** is a single target.
- If a search returns known targets, either by lookup or locate, then identify returns their characteristics
- Conversely, if a search returns targets matching particular characteristics, either by *browse* or *explore*, then identify returns specific references

Compare



- The scope of **compare** is multiple targets.
- Comparison tasks are typically more difficult than identify tasks and require more sophisticated idioms to support the user.

Summarize



- The scope of **summarize** task is all possible targets.
- A synonym for *summarize* is **overview**, a term is frequently used in the vis literature both as a verb, where it means to provide a comprehensive view of everything, and as a noun, where it means a summary display of everything.
- The goal of providing an overview is extremely common in visualization



Targets



Actions

Analyze
Produce
Search
Query

Targets

How

Analyzing and Deriving: Examples

Comparing Two Idioms
Deriving One Attribute
Deriving Many New
Attributes

Why: Targets

Targets

→ All Data

→ Trends



→ Outliers



→ Features



→ Attributes

→ One

→ Distribution



→ Extremes



→ Many

→ Dependency



→ Correlation



→ Similarity



→ Network Data

→ Topology



→ Paths



→ Spatial Data

→ Shape



Targets



- The actions discussed above refer to a **target**, meaning some aspect of the data that is of interest to the user.
- Targets are *nouns*, whereas actions are *verbs*.
- The idea of a target is explicit with *search* and *query* actions.
- It is more implicit with the use actions, but still relevant, the thing that the user *presents* or *discovers*

All data



- Three high-level targets are very broadly relevant, for all kinds of data: *trends*, *outliers*, and *features*.
- A **trend** is a high-level characterization of a pattern in the data. Simple examples of trends include increases, decreases, peaks, troughs, and plateaus.
- Almost inevitably, some data doesn't fit well with that backdrop; those elements are the **outliers**.
- The exact definition of **features** is task dependent, meaning any particular structures of interest.

Attributes



Attributes are specific properties that are visually encoded.

- One attribute
 - The lowest-level target for an attribute is to find an individual value.
 - Another frequent target of interest is to find the extremes.
 - A very common target that has high-level scope is the distribution of all values for an attribute
- Some targets encompass the scope of multiple attributes: **dependencies**, **correlations**, and **similarities** between attributes.

Network & spatial data



- Network data specifies relationships between nodes as links.
 - The fundamental target with network data is to understand the structure of these interconnections; that is, the network's **topology**.
 - A more specific topological target is a **path** of one or more links that connects two nodes.
- For spatial data, understanding and comparing the geometric **shape** is the common target of user actions.

How





Actions

Analyze
Produce
Search
Query

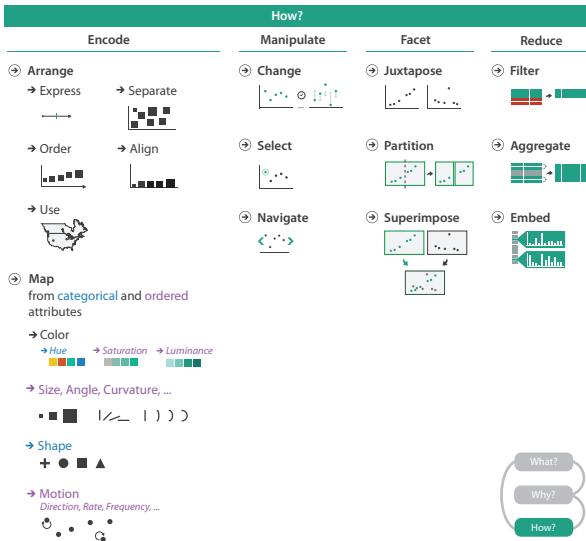
Targets

How

Analyzing and Deriving: Examples

Comparing Two Idioms
Deriving One Attribute
Deriving Many New Attributes

The Big Picture





Analyzing and Deriving: Examples

- Comparing Two Idioms
- Deriving One Attribute
- Deriving Many New Attributes

Comparing Two Idioms

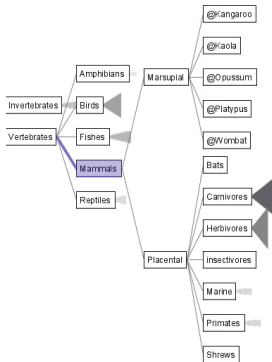


- The **what–why–how** analysis framework is useful for comparative analysis, for example, to examine two different vis tools that have different answers for the question of how the idiom is designed when used for exactly the same context of why and what at the abstraction level.



Example

- Comparing two idioms. (a) SpaceTree (b) TreeJuxtaposer.



(a)



(b)



Actions

Analyze
Produce
Search
Query

Targets

How

Analyzing and Deriving: Examples

Comparing Two Idioms

Deriving One Attribute

Deriving Many New
Attributes

Analysis

What?

→ Tree



Why?

→ Actions

→ Present → Locate → Identify



→ Targets

→ Path between two nodes



How?

→ SpaceTree

→ Encode → Navigate → Select → Filter → Aggregate



→ TreeJuxtaposer

→ Encode → Navigate → Select → Arrange





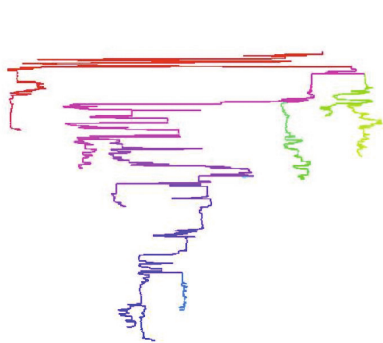
Deriving One Attribute

- In a vis showing a complex network or tree, it is useful to be able to filter out most of the complexity by drawing a simpler picture that communicates the key aspects of its topological structure.
- One way to support this kind of summarization is to calculate a new derived attribute that measures the importance of each node in the graph and filter based on that attribute.
- Many different approaches to calculating importance have been proposed; **centrality metrics** do so in a way that takes into account network topology

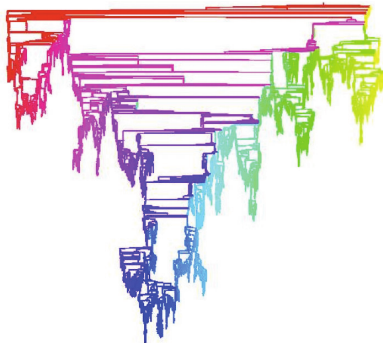


Example

- The derived quantitative attribute of Strahler numbers is used to filter the tree in order to create a recognizable summary. (a) The important skeleton of a large tree is visible when only 5000 of the highest-ranked nodes are drawn. (b) The full tree has over a half million nodes



(a)



(b)



Actions

Analyze
Produce
Search
Query

Targets

How

Analyzing and Deriving: Examples

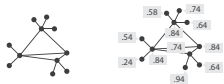
Comparing Two Idioms

Deriving One Attribute

Deriving Many New
Attributes

Analysis

Task 1



In Tree → **Out** Quantitative attribute on nodes

What?

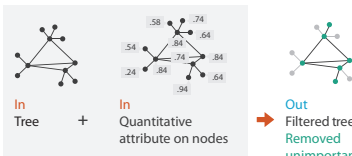
→ In Tree

→ Out Quantitative attribute on nodes

Why?

→ Derive

Task 2



In Tree + **In** Quantitative attribute on nodes → **Out** Filtered tree
Removed unimportant parts

What?

→ In Tree

→ In Quantitative attribute on nodes

→ Out Filtered tree

Why?

→ Summarize

→ Topology

How?

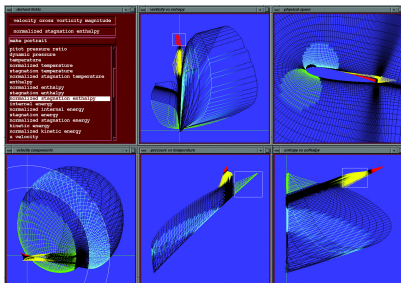
→ Reduce

→ Filter



Example

- Data transformations can shed light into spatial data as well. In an example from computational fluid dynamics, linked derived spaces are used for feature detection.
- Computational fluid dynamics vis showing the list of many derived attributes (top left), one view of the original spatial field (top right), and four other views showing pairs of selected derived attributes. The multiple juxtaposed views are coordinated with shared colored highlights.





Actions

Analyze
Produce
Search
Query

Targets

How

Analyzing and Deriving: Examples

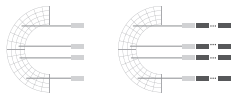
Comparing Two Idioms

Deriving One Attribute

Deriving Many New Attributes

Analysis

Task 1



In Spatial field → **Out** Many quantitative attributes

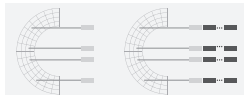
What?

- **In** Spatial field
- **Out** Many quantitative attributes

Why?

- Derive

Task 2



In Spatial field + **In** Many quantitative attributes → **Out** Juxtaposed attribute plots with linked coloring



What?

- **In** Spatial field
- **In** Many quantitative attributes
- **Out** Juxtaposed attribute plots with linked coloring

Why?

- **Actions**
 - Discover
 - Explore
 - Browse
 - Identify
 - Compare
- **Targets**
 - Features

How?

- **Map**
 - Hue
- **Facet**
 - Juxtapose
 - Partition
- **Arrange**
 - Express
- **Manipulate**
 - Select
 - Navigate

References



Goodfellow, I., Bengio, Y., and Courville, A. (2016).

Deep learning.

MIT press.



Munzner, T. (2014).

Visualization analysis and design.

CRC press.



Russell, S. and Norvig, P. (2016).

Artificial intelligence: a modern approach.

Pearson Education Limited.



Ward, M. O., Grinstein, G., and Keim, D. (2015).

Interactive data visualization: foundations, techniques, and applications.

CRC Press.