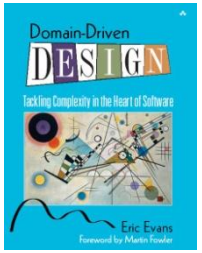
Domain-Driven Design

Useful Models for Complex Problems

Eric Evans
domainlanguage.com

domain language



Why bother with models?

‘We should do a nice design, but we just don’t have time.’

Religion

‘Modeling and design take extra time, but they pay off in the long run.’

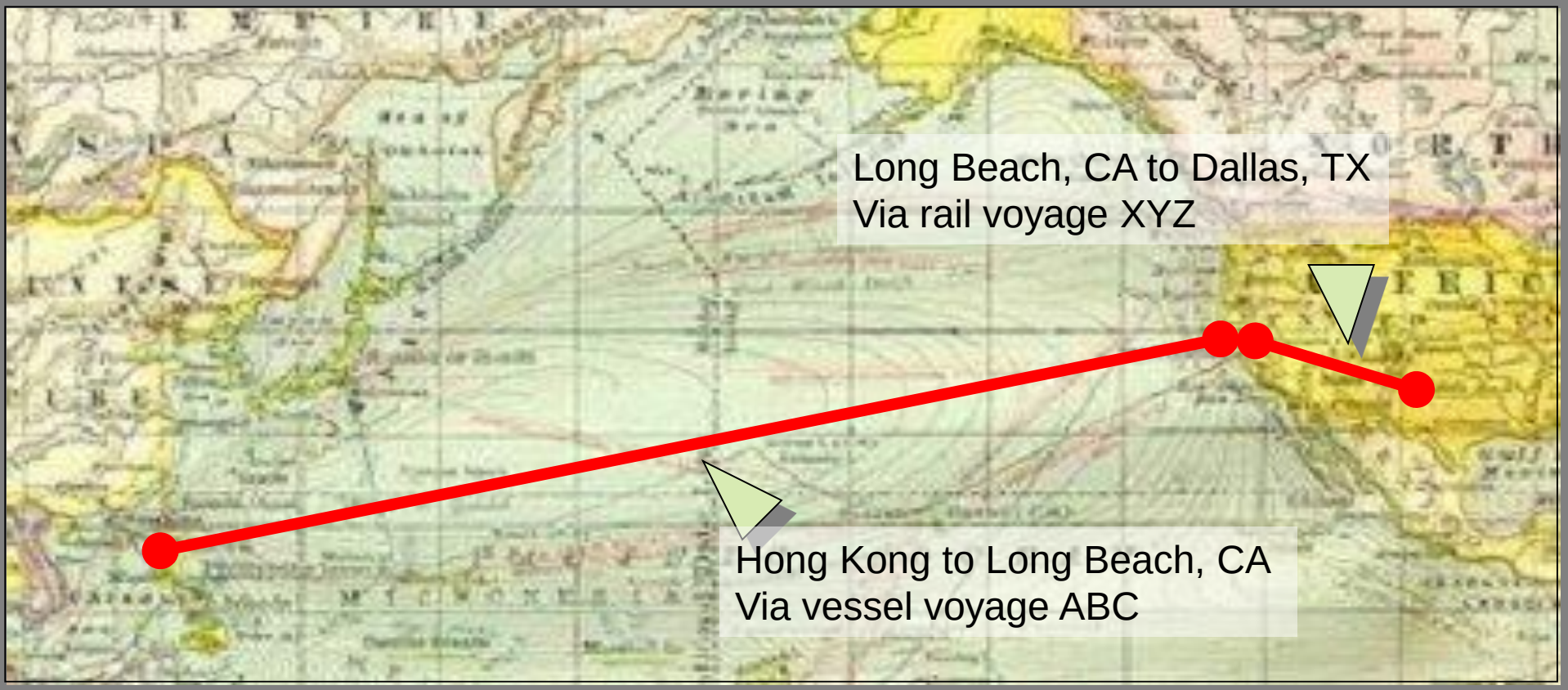
Speculation

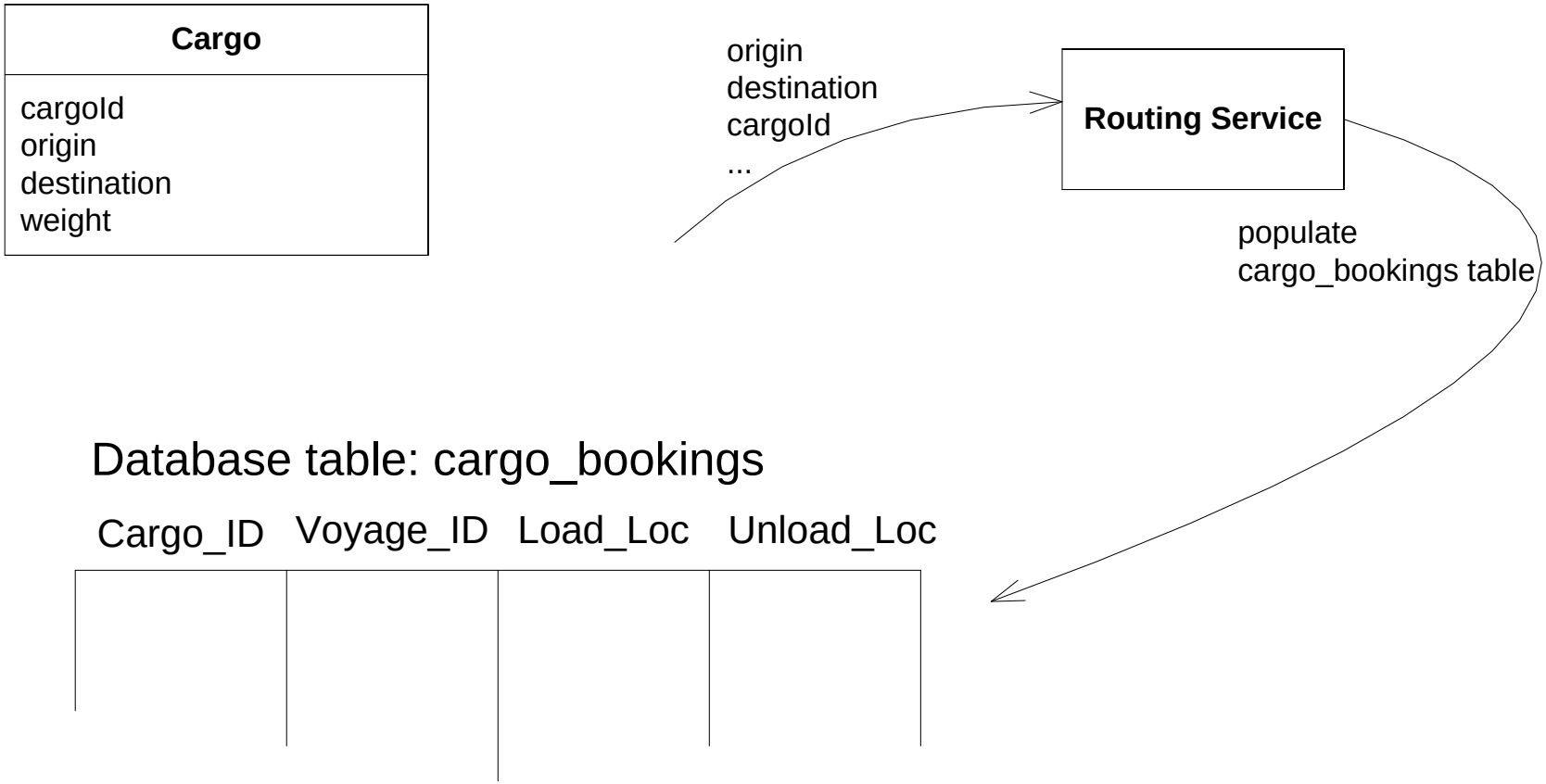
Modeling and design are often *the quickest path to the actual goal*.

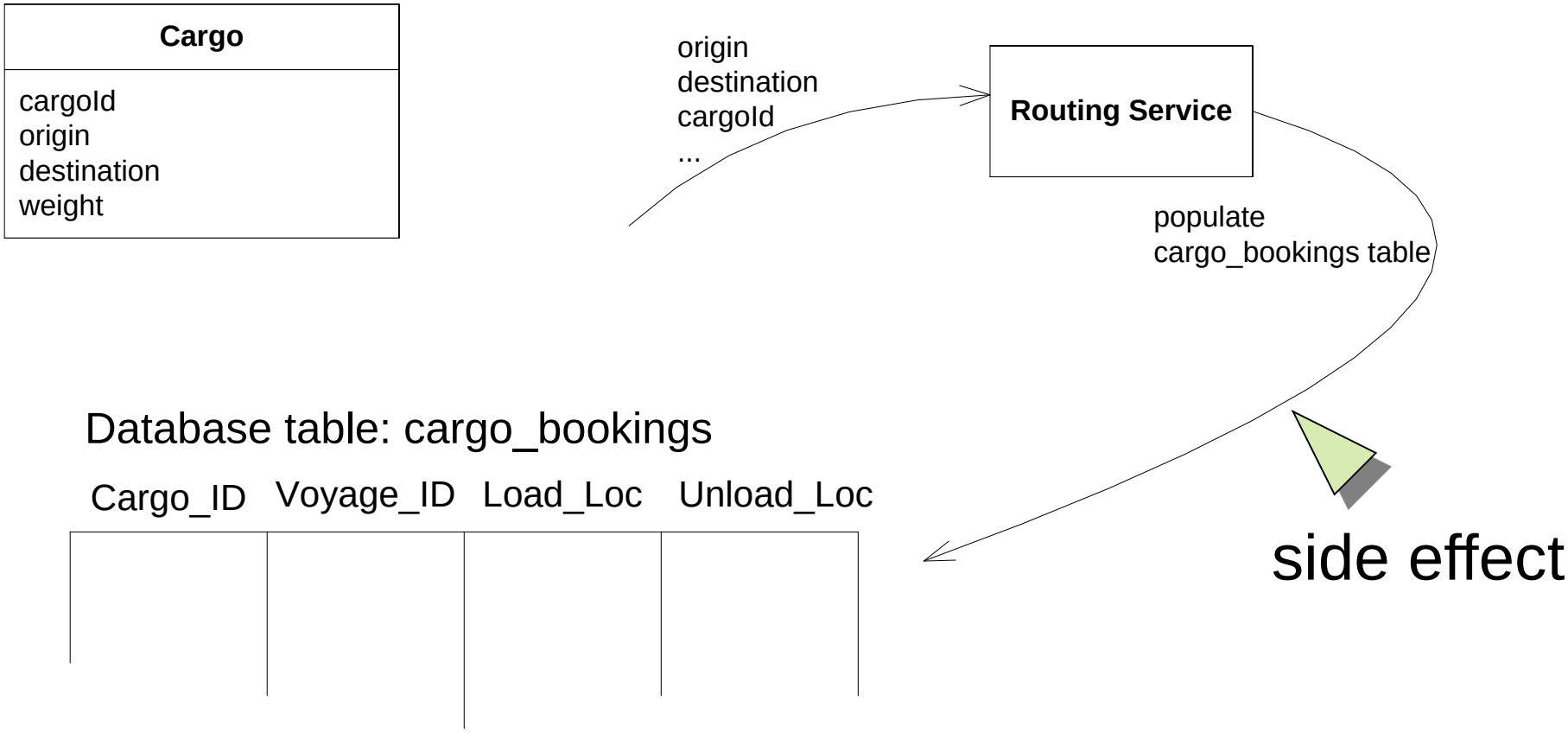
What is your goal?

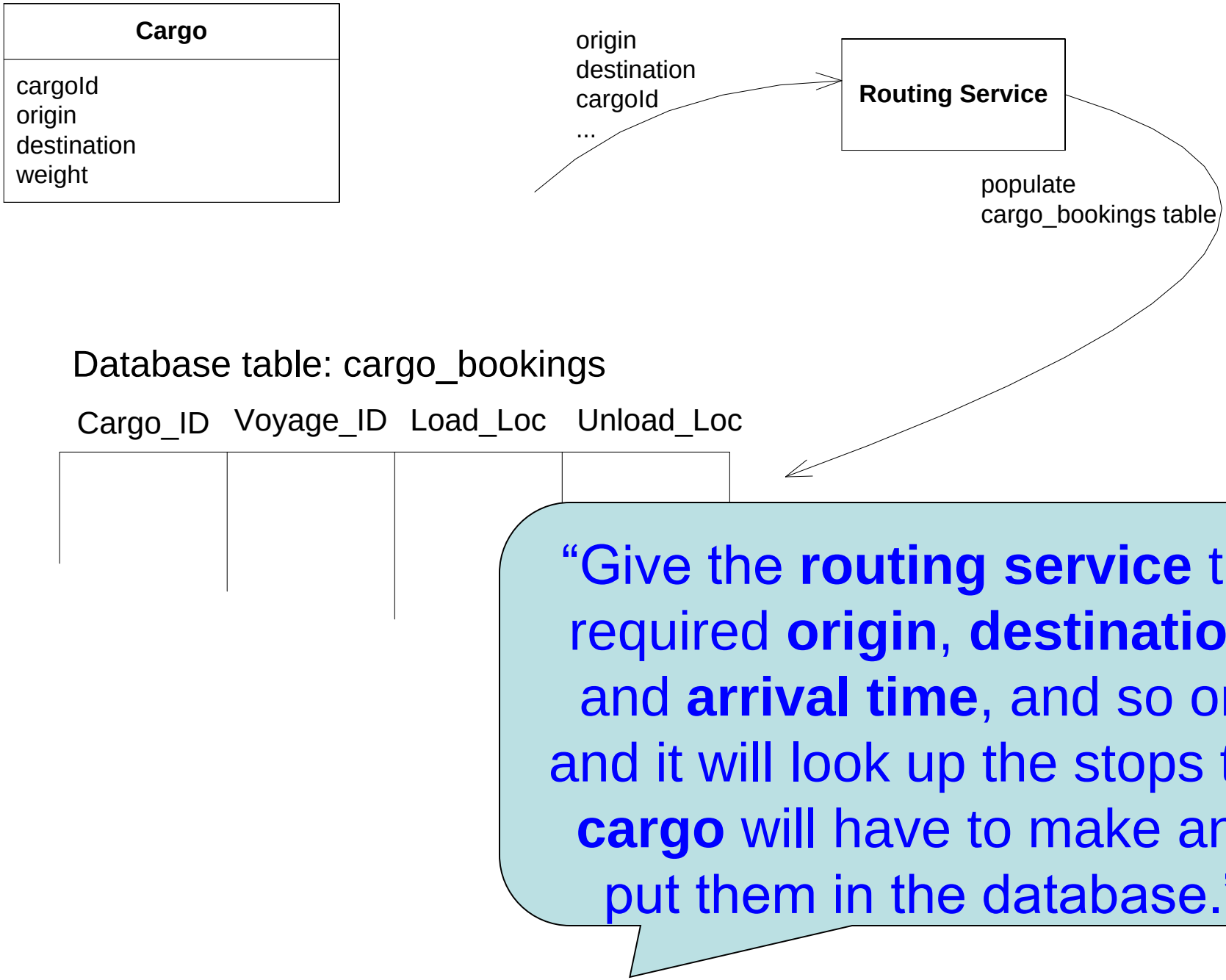
- Implement this feature?
- Complete a releasable set of stories with an acceptable level of bugs?
- Deliver a release that the team can continue to extend in the next release?
- Deliver a clear and cohesive user experience?

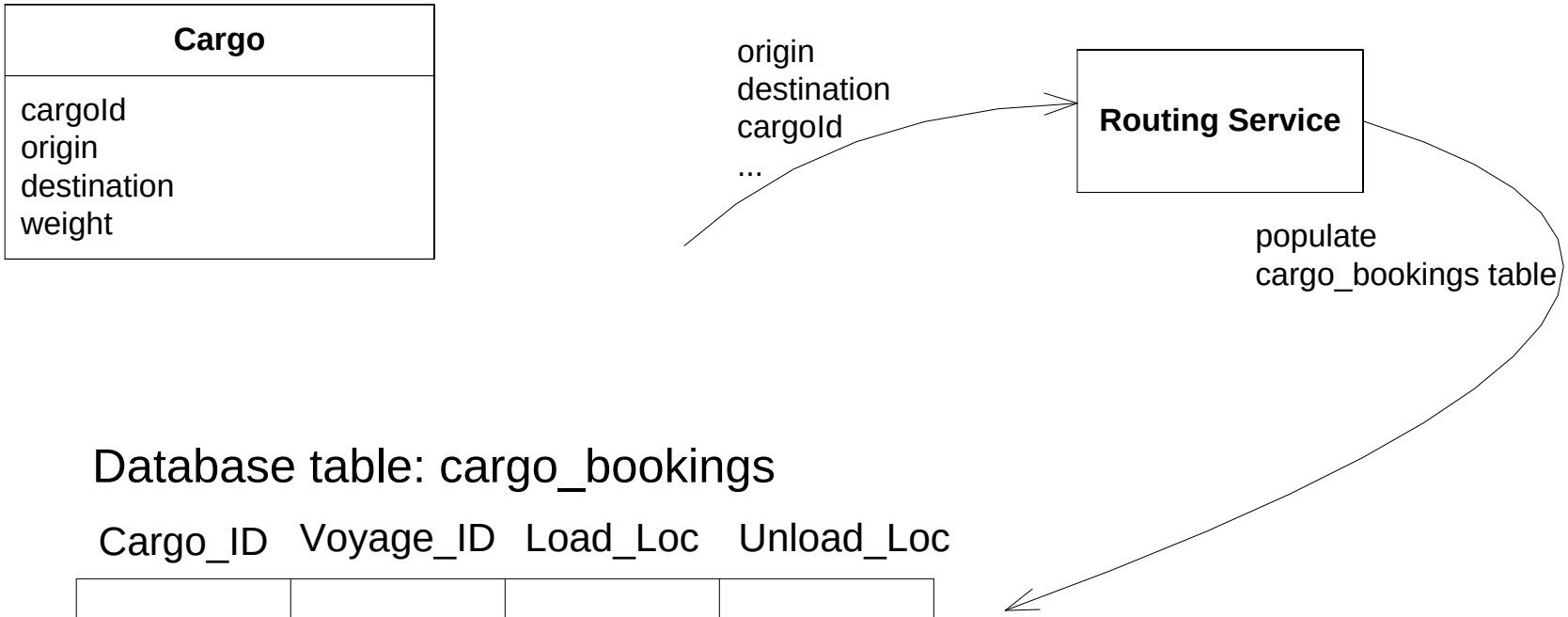
The critical complexity of most software projects is in understanding the domain itself.


Long Beach, CA to Dallas, TX
Via rail voyage XYZ
Hong Kong to Long Beach, CA
Via vessel voyage ABC


Cargo
cargoId
origin
destination
weight
origin
destination
cargoId
...
Routing Service
populate
cargo_bookings table
Database table: cargo_bookings
Cargo_ID
Voyage_ID
Load_Loc
Unload_Loc


Cargo
cargoId
origin
destination
weight
origin
destination
cargoId
...
Routing Service
populate
cargo_bookings table
Database table: cargo_bookings
Cargo_ID
Voyage_ID
Load_Loc
Unload_Loc
side effect


Cargo
cargoId
origin
destination
weight
origin
destination
cargoId
...
Routing Service
populate
cargo_bookings table
Database table: cargo_bookings
Cargo_ID
Voyage_ID
Load_Loc
Unload_Loc
“Give the routing service the required origin, destination, and arrival time, and so on, and it will look up the stops the cargo will have to make and put them in the database.”

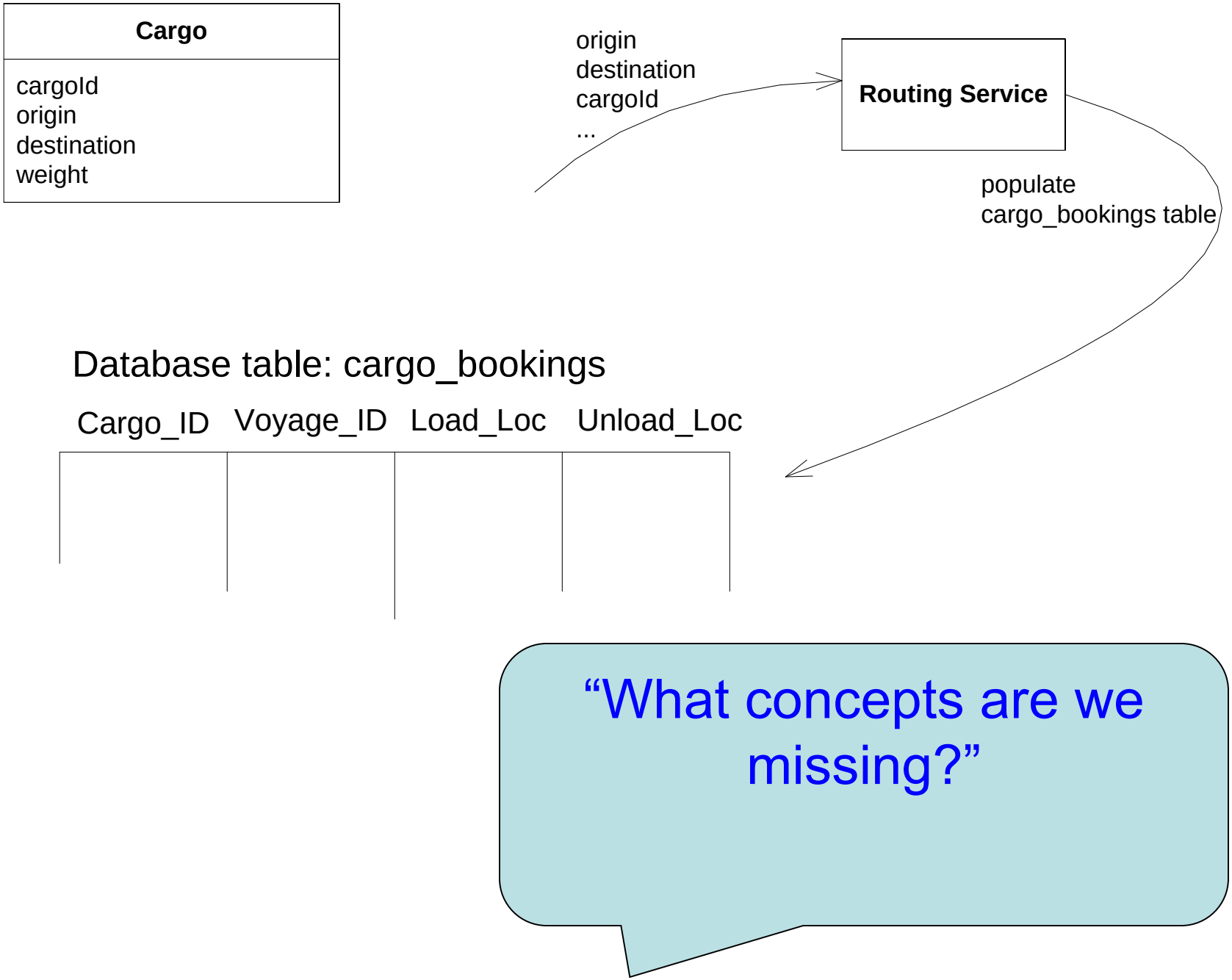


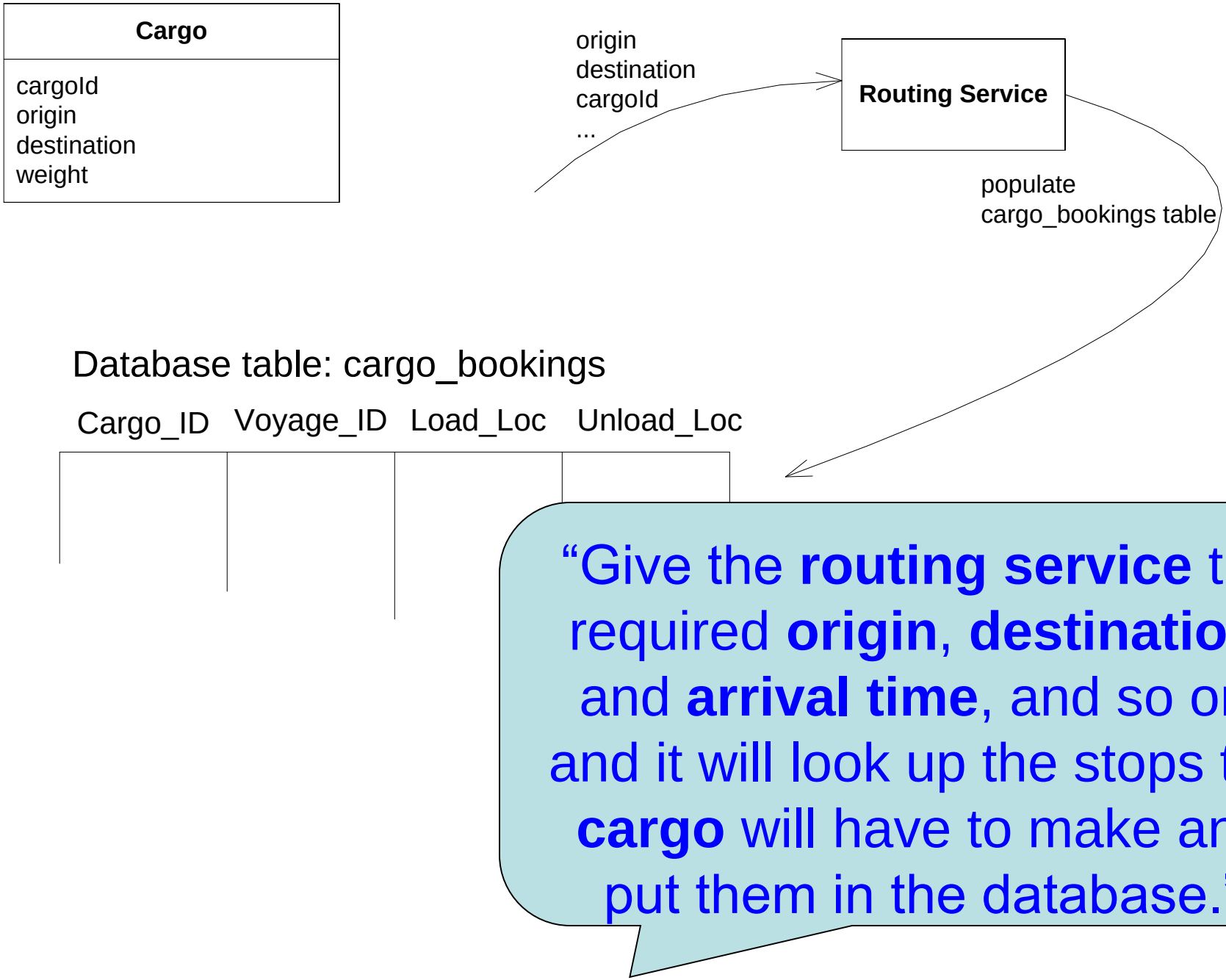
Database table: cargo_bookings

Cargo_ID	Voyage_ID	Load_Loc	Unload_Loc

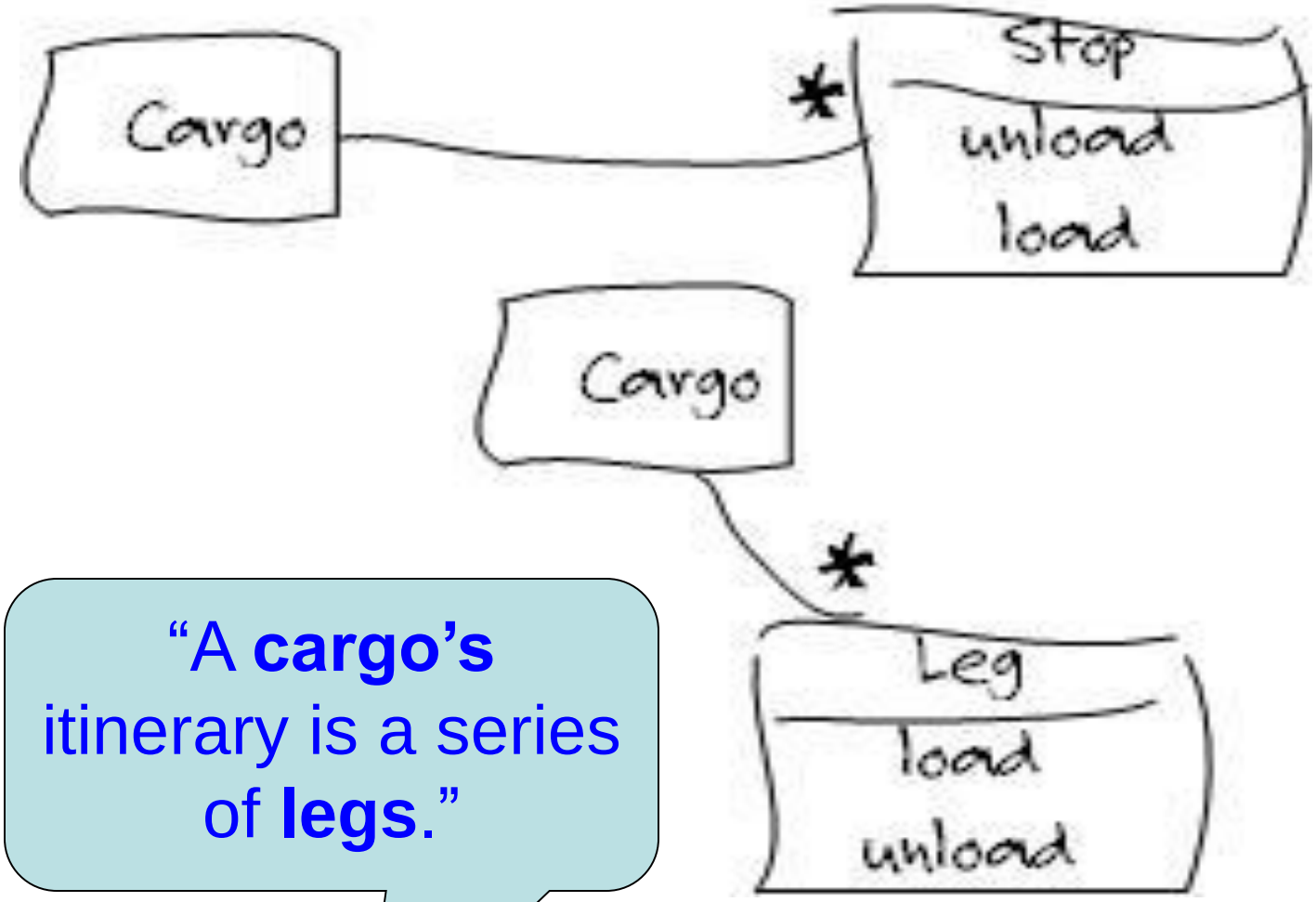
"Each table row shows load and unload."

"Give the **routing service** the required **origin**, **destination**, and **arrival time**, and so on, and it will look up the stops the **cargo** will have to make and put them in the database."


Cargo
cargoId
origin
destination
weight
origin
destination
cargoId
...
Routing Service
populate
cargo_bookings table
Database table: cargo_bookings
Cargo_ID
Voyage_ID
Load_Loc
Unload_Loc
“What concepts are we missing?”


Cargo
cargoId
origin
destination
weight
origin
destination
cargoId
...
Routing Service
populate
cargo_bookings table
Database table: cargo_bookings
Cargo_ID
Voyage_ID
Load_Loc
Unload_Loc
“Give the routing service the required origin, destination, and arrival time, and so on, and it will look up the stops the cargo will have to make and put them in the database.”

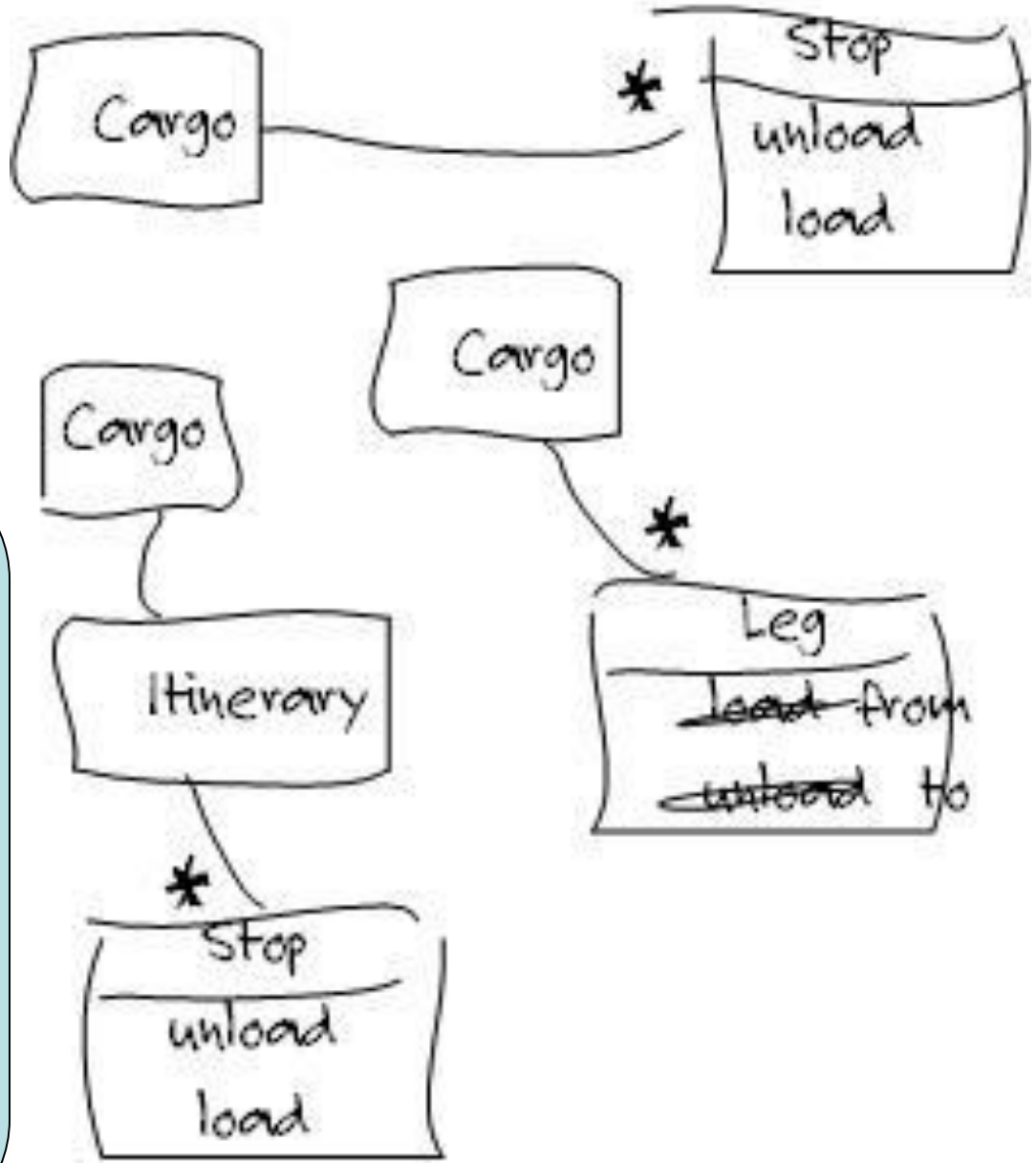
"A **cargo** makes a series of **stops**. At each **stop** it is **unloaded** from a transport and **loaded** onto another."



"A **cargo's** itinerary is a series of **legs**."

"The **cargo** is **loaded** onto a transport at the beginning of a **leg** and **unloaded** at the end."

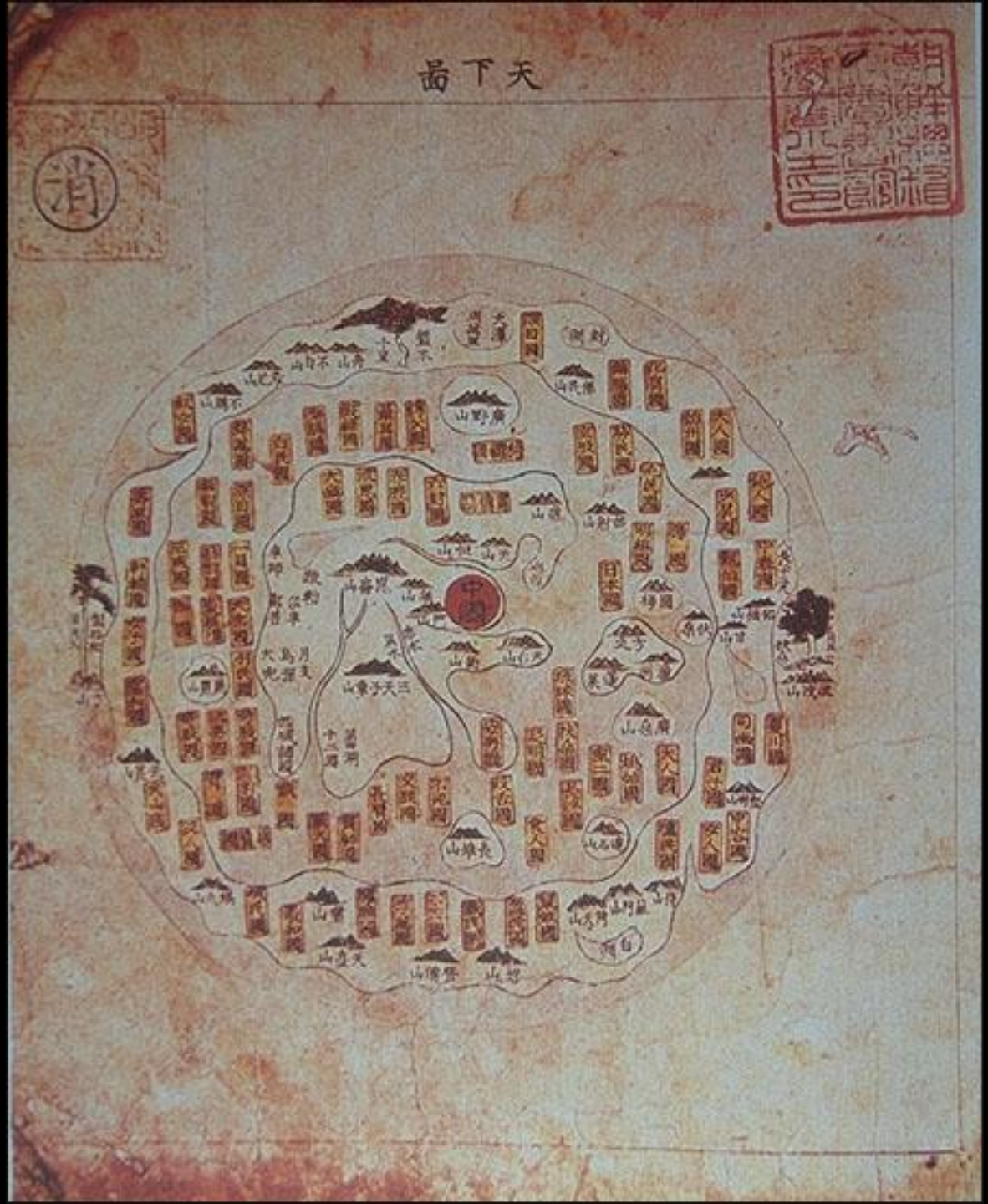
"A **cargo's itinerary** gives instructions for each **stop**, to **unload** and **load** onto a different transport."

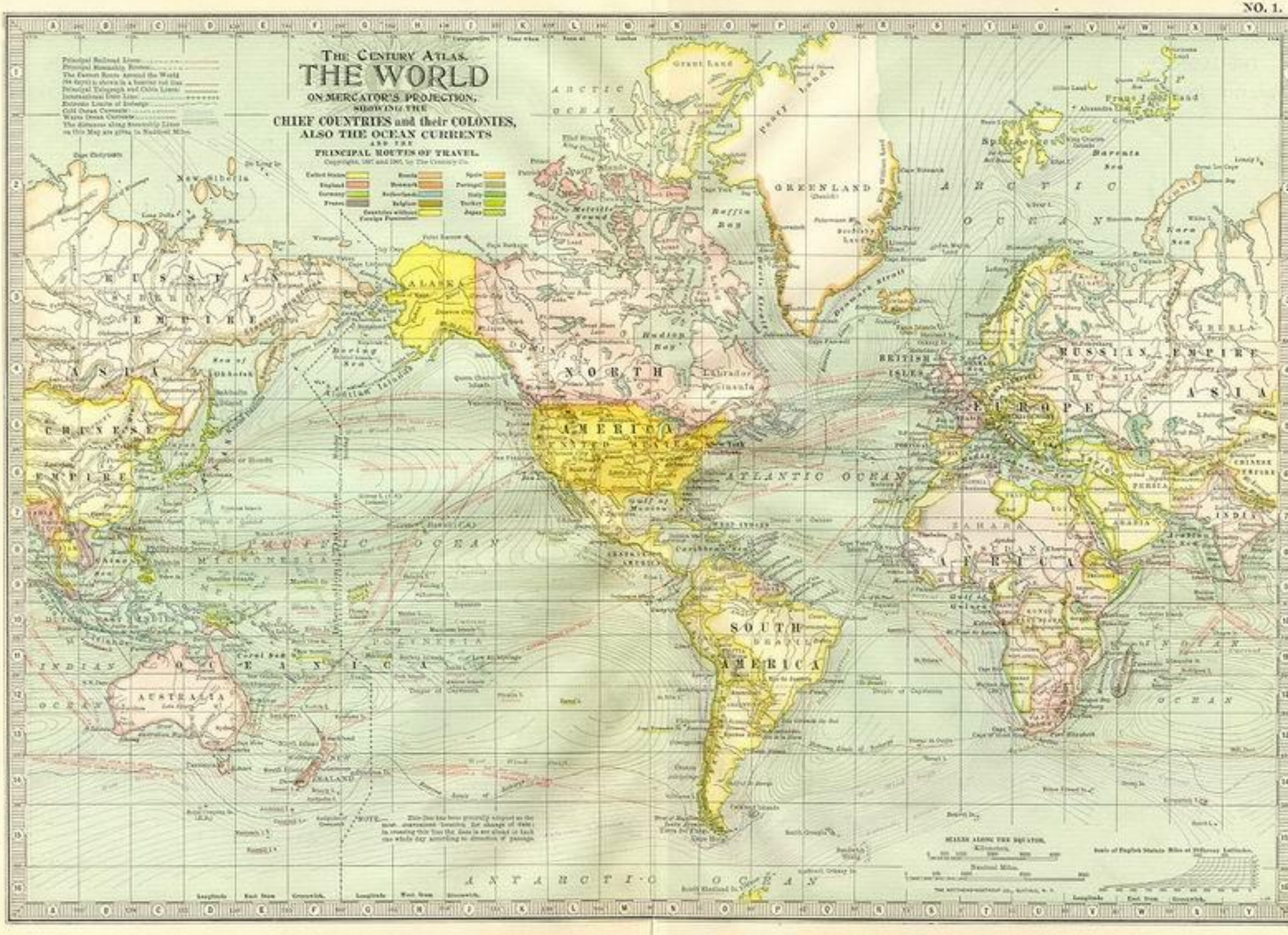


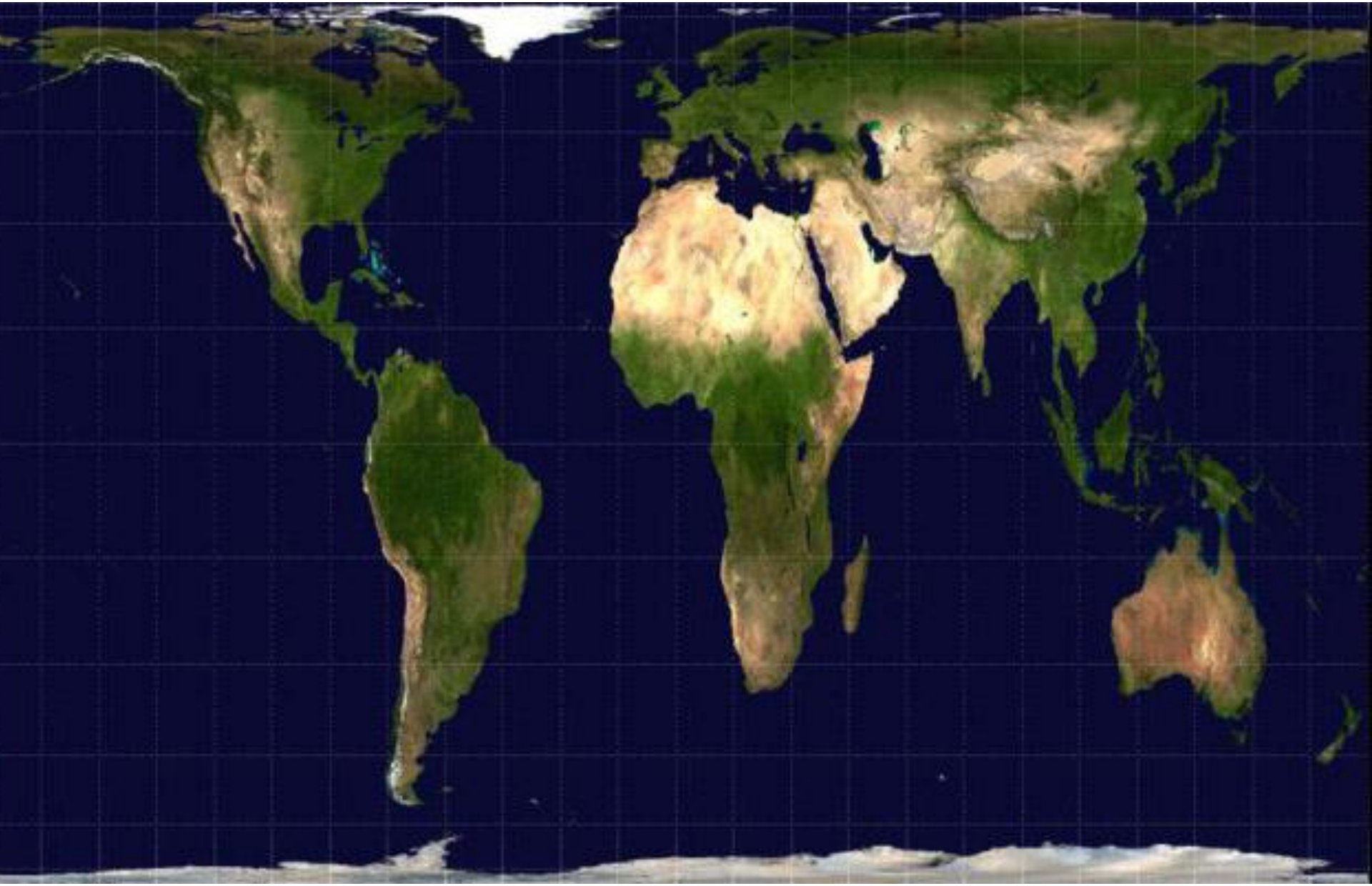
Stops or Legs?

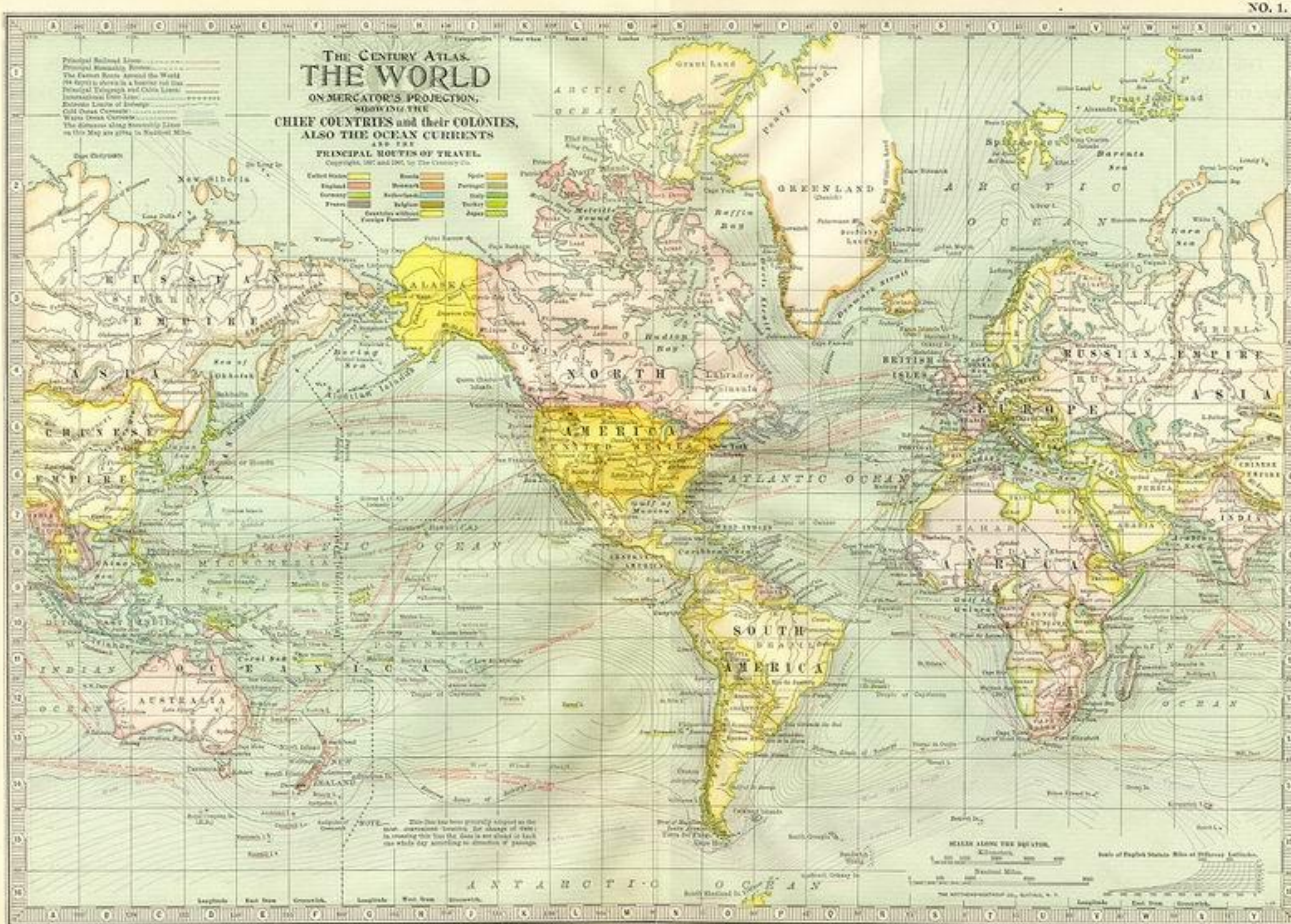
Which is better?

What is a model?


天下畐


THE CENTURY ATLAS.
THE WORLD
ON MERCATOR'S PROJECTION,
SHOWING THE
CHIEF COUNTRIES and their COLONIES,
ALSO THE OCEAN CURRENTS
AND THE
PRINCIPAL ROUTES OF TRAVEL.
ARCTIC OCEAN
GREENLAND
ALASKA
NORTH
AMERICA
UNITED STATES
ATLANTIC OCEAN
PACIFIC OCEAN
SOUTH
AMERICA
EUROPE
RUSSIAN EMPIRE
ASIA
AFRICA
SAHARA
INDIA
CHINESE EMPIRE
AUSTRALIA
OCEANICA
INDIAN OCEAN
ANTARCTIC OCEAN




The Century Atlas.
THE WORLD
ON MERCATOR'S PROJECTION,
SHOWING THE
CHIEF COUNTRIES and their COLONIES,
ALSO THE OCEAN CURRENTS
AND THE
PRINCIPAL ROUTES OF TRAVEL.
ARCTIC OCEAN
GREENLAND
ALASKA
NORTH
AMERICA
UNITED STATES
ATLANTIC OCEAN
PACIFIC OCEAN
SOUTH
AMERICA
AFRICA
EUROPE
RUSSIAN EMPIRE
ASIA
CHINESE
EMPIRE
INDIA
AUSTRALIA
OCEANICA
INDIAN
OCEAN
ANTARCTIC OCEAN

Some statements about this model

- The Earth is approximated as a sphere, with points of interest located on the surface.
- There are functions that map points on the surface of a sphere onto a cylinder (projections).
- Mercator’s Projection preserves angles / direction of points relative to each other.


NOVA ET AVCTA ORBIS TERRAE DESCRIPTIO AD VSVM NA

Defining Our Terms

domain A sphere of knowledge or activity.

model A system of abstractions representing selected aspects of the domain.

A *model* is a distilled form of *domain* knowledge, assumptions, rules and choices.

It’s not about “realism”

A model serves some *use*.

Usefulness is specific to *particular scenarios*.

There are always *multiple models* of the domain.

Stops or Legs?

Which is more *useful*?

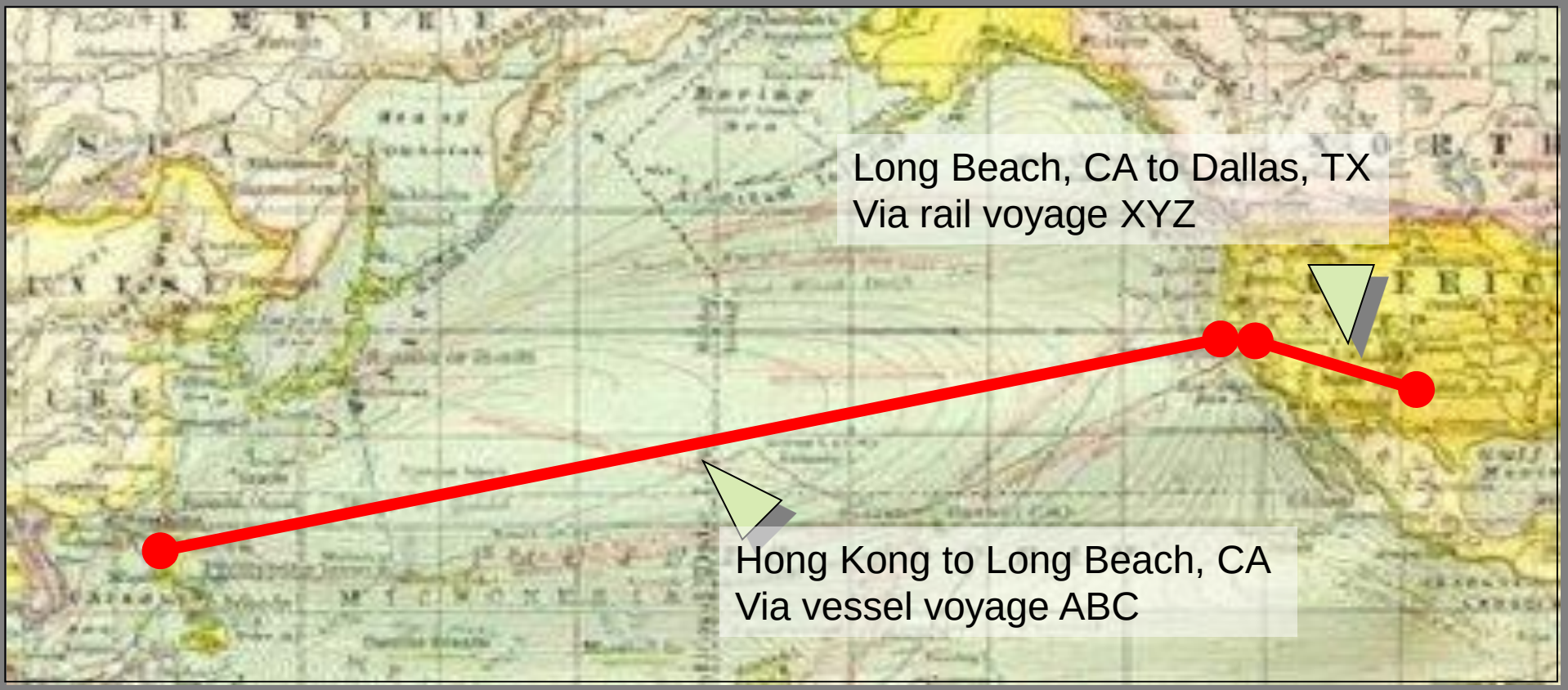
Useful *for what*?

Shipping Reference Scenarios

- Route and book a new shipment

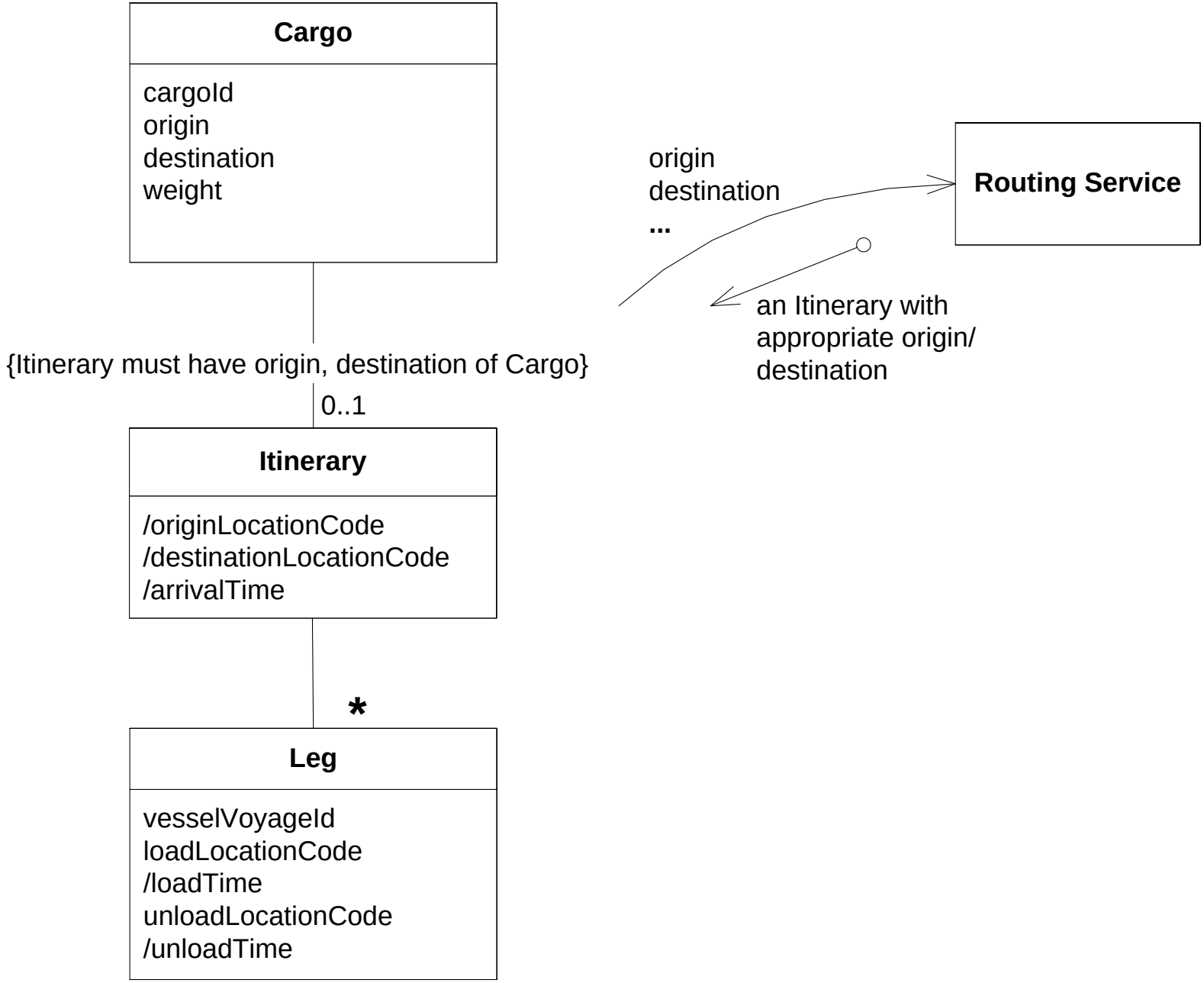
Shipping Reference Scenarios

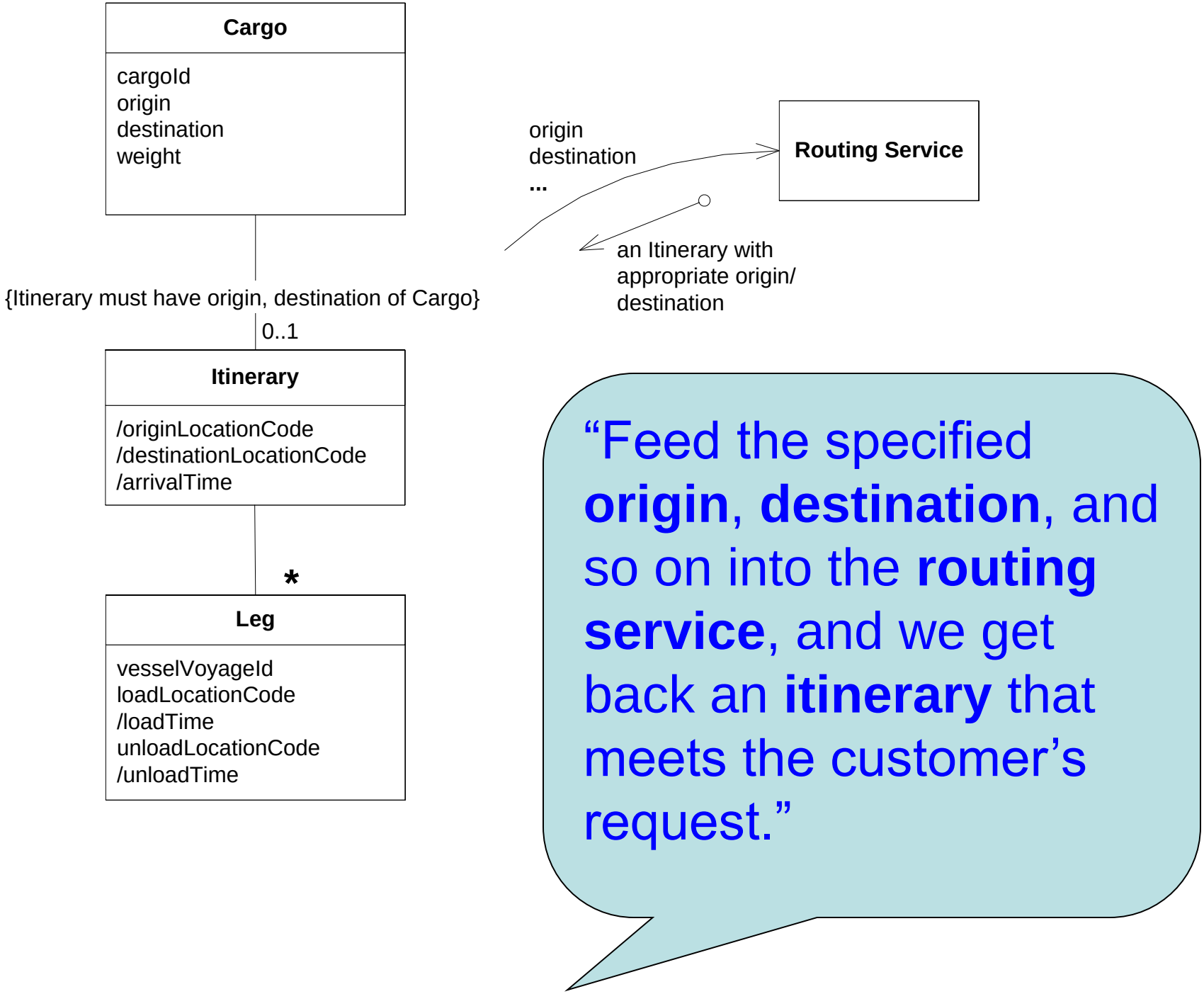
- Route and book a new shipment
- **Reroute mid-transit at customer order**

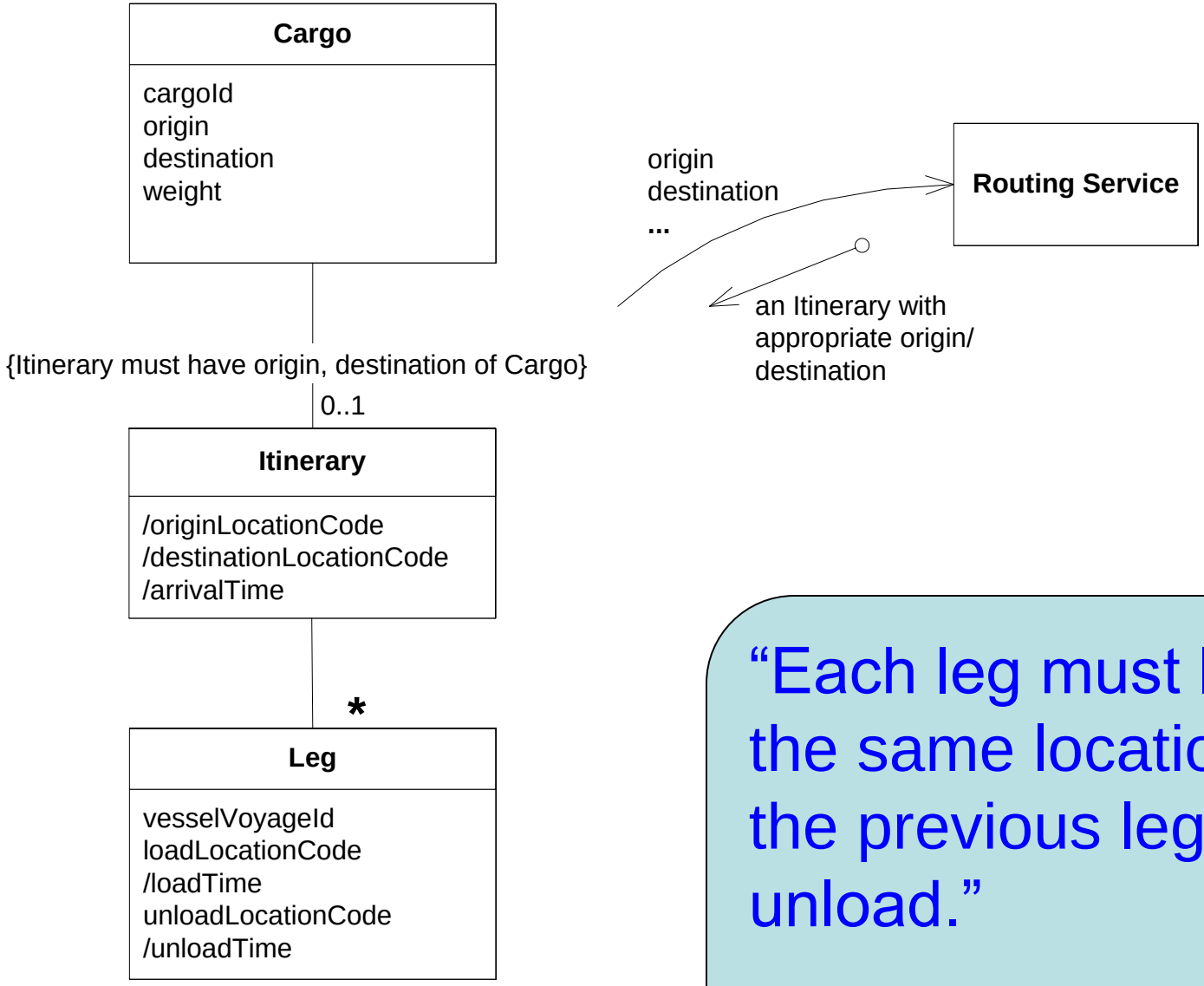

Long Beach, CA to Dallas, TX
Via rail voyage XYZ
Hong Kong to Long Beach, CA
Via vessel voyage ABC


Hong Kong to Seattle, WA
Via vessel voyage DEF


Long Beach, CA to Seattle, WA
Via rail voyage UVW
Hong Kong to Long Beach, CA
Via vessel voyage ABC


Cargo
cargoId
origin
destination
weight
origin
destination
...
Routing Service
an Itinerary with
appropriate origin/
destination
{Itinerary must have origin, destination of Cargo}
0..1
Itinerary
/originLocationCode
/destinationLocationCode
/arrivalTime
*
Leg
vesselVoyageId
loadLocationCode
/loadTime
unloadLocationCode
/unloadTime


Cargo
cargoId
origin
destination
weight
origin
destination
...
Routing Service
an Itinerary with
appropriate origin/
destination
{Itinerary must have origin, destination of Cargo}
0..1
Itinerary
/originLocationCode
/destinationLocationCode
/arrivalTime
*
Leg
vesselVoyageId
loadLocationCode
/loadTime
unloadLocationCode
/unloadTime
"Feed the specified origin, destination, and so on into the routing service, and we get back an itinerary that meets the customer's request."



"Each leg must load in the same location as the previous legs unload."

Yes, But What If…?

- Route and book a new shipment.
- **Before each voyage arrival, issue purchase orders to vendors in the port for unloading, loading, storage and routing within the terminal.**

‘We have to get the model right first, before we write the code.’

Non Sequitur

Up Front Analysis Locks in Ignorance

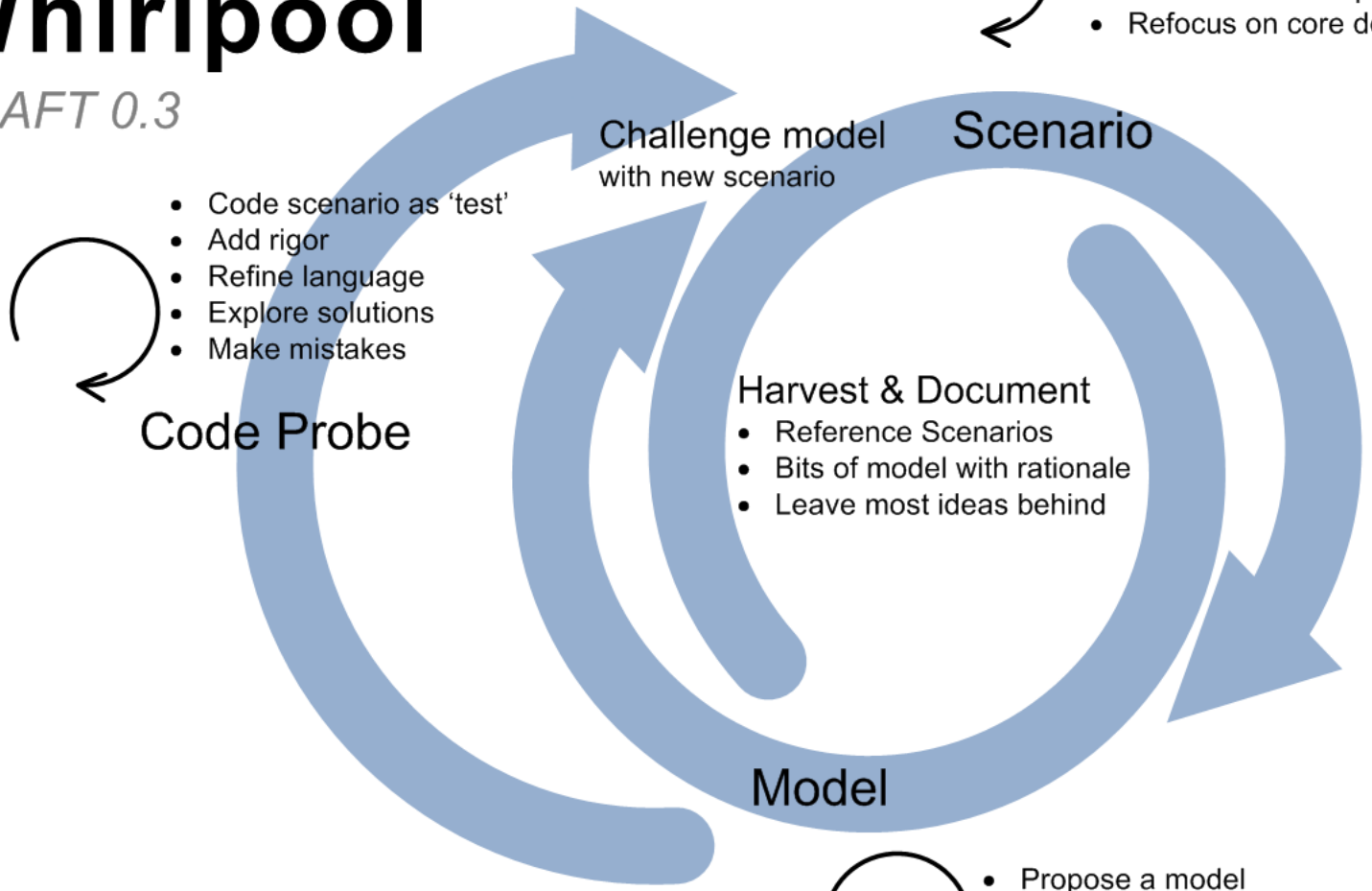
Models are distilled knowledge.

At the beginning of a project, the team is as ignorant as it will ever be.

Model Exploration

Whirlpool

DRAFT 0.3



- Tell us a story.
- Flesh it out.
- Refocus on hard part.
- Refocus on core domain

- Code scenario as ‘test’
- Add rigor
- Refine language
- Explore solutions
- Make mistakes

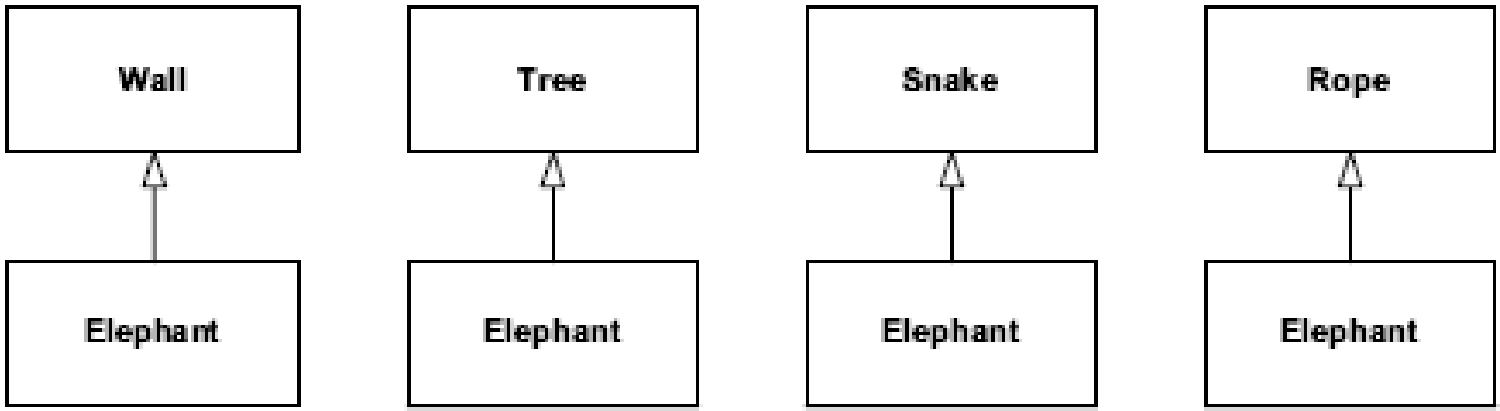
- Propose a model
- Walkthrough states
- Walkthrough solutions
- Explore language
- Make mistakes

domain language

www.domainlanguage.com/ddd/whirlpool

There are always multiple models.

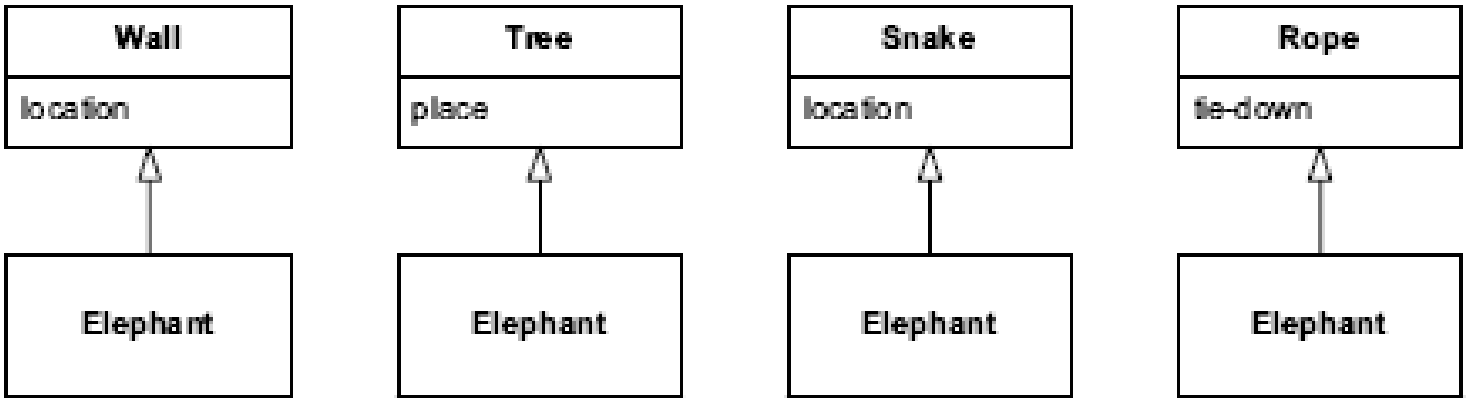
Blind Men and Elephant


Wall
Tree
Snake
Rope
Elephant
Elephant
Elephant
Elephant

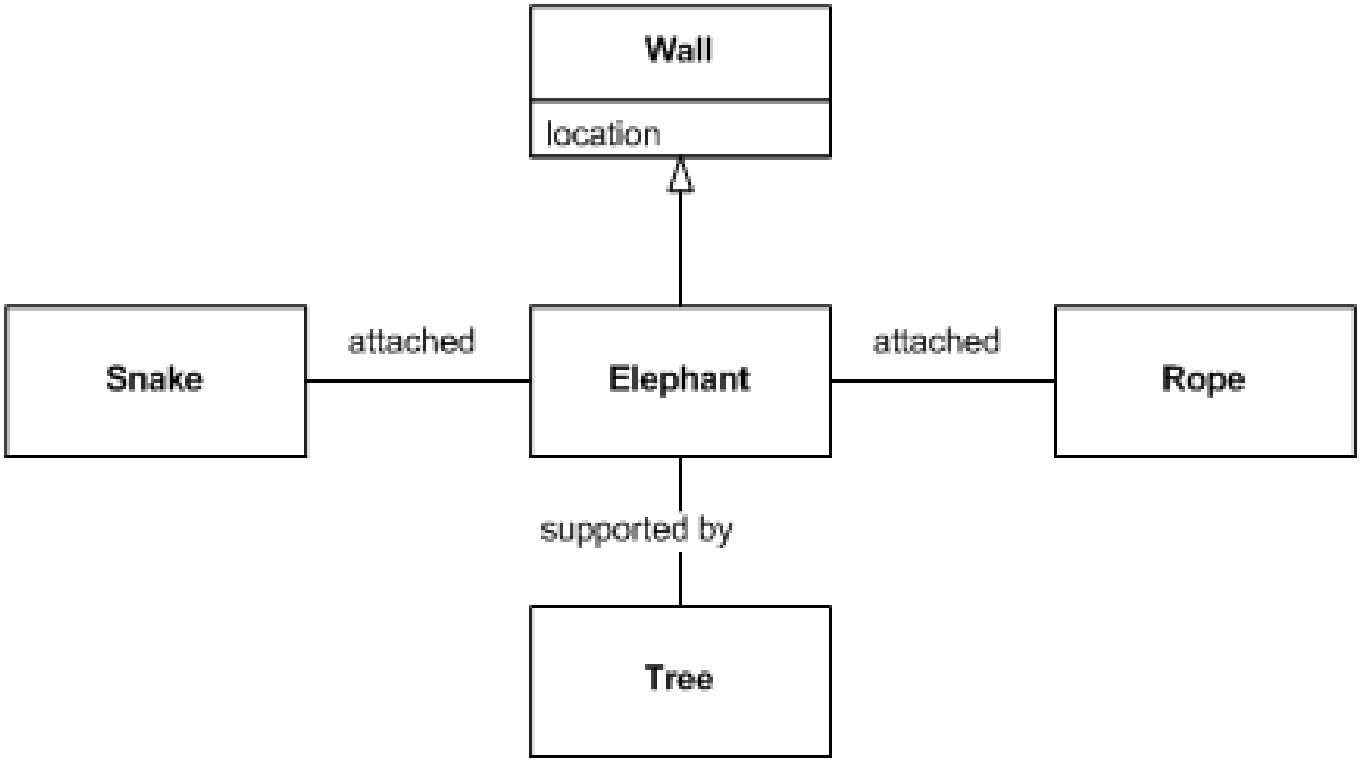
Two Modelling Mistakes

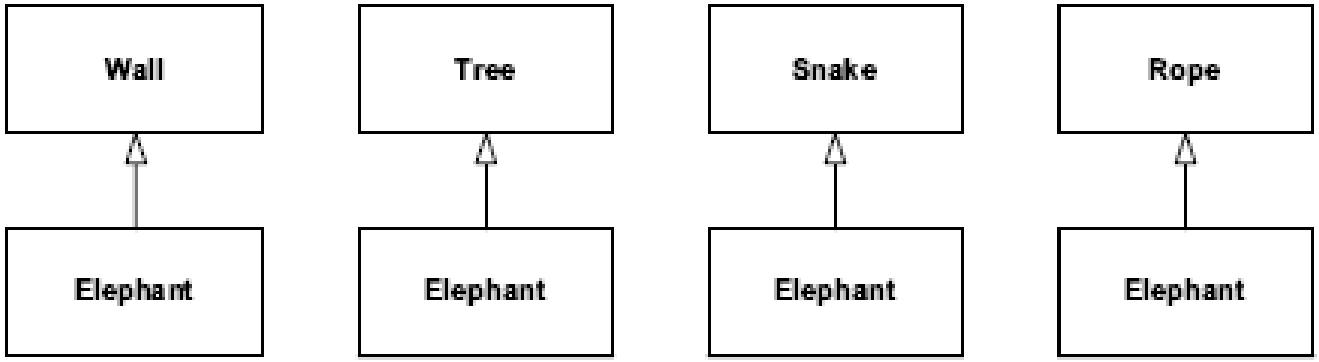
- They keep trying to figure out the “true” nature of an elephant.
- **They insist on just one model, in spite of incomplete information and possibly different goals.**

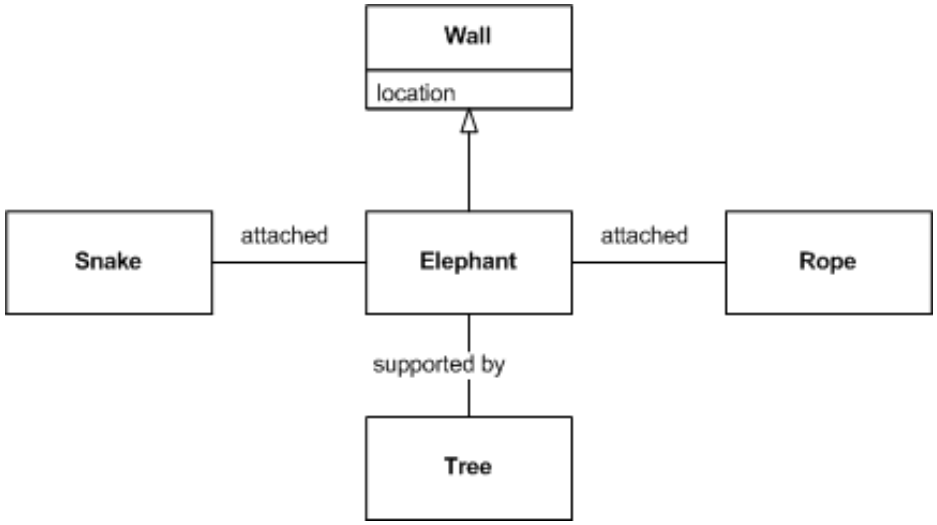
Bounded Context An operational definition of where a particular model is well-defined and applicable. (Typically a subsystem, or the work owned by a particular team).

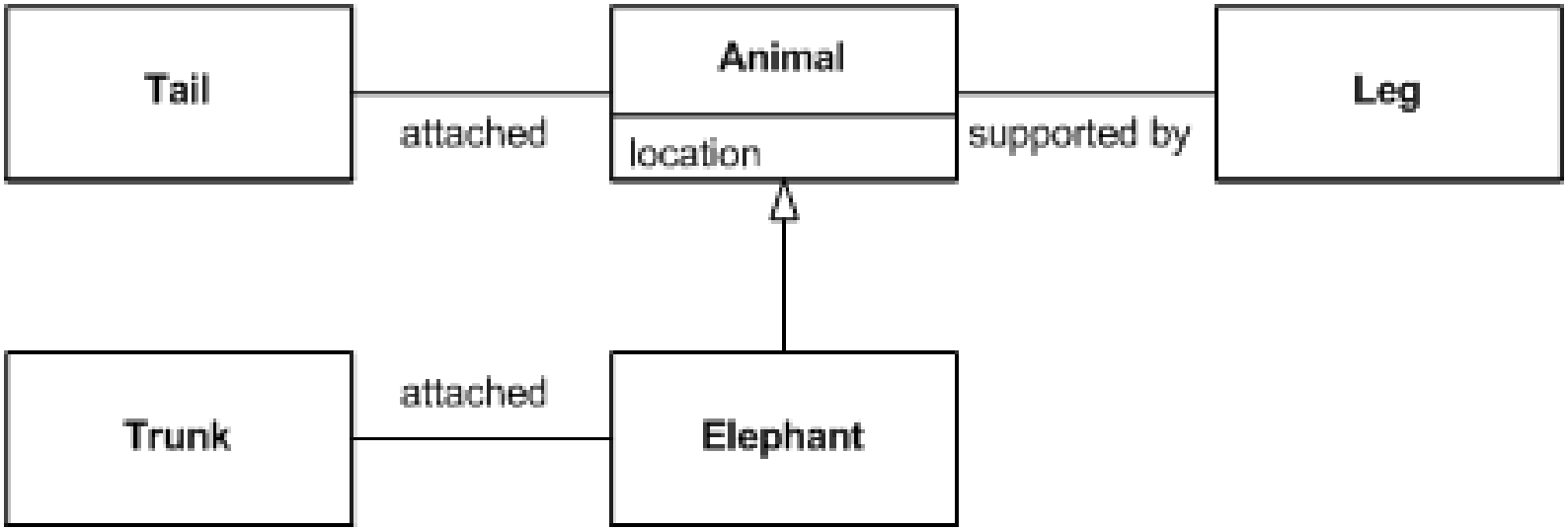


Translations: {Wall.location ↔ Tree.place ↔ Snake.location ↔ Rope.tie-down}


Wall
location
Snake
attached
Elephant
attached
Rope
supported by
Tree


Wall
Tree
Snake
Rope
Elephant
Elephant
Elephant
Elephant


Wall
location
Snake
attached
Elephant
attached
Rope
supported by
Tree


Tail
attached
Animal
location
supported by
Leg
Trunk
attached
Elephant

Precision designs are fragile.

They require a bounded context containing a unified model.

Not all of a large system will be well designed.

Where to look for more

- Model Exploration
  - domainlanguage.com/ddd/whirlpool
  - Section 3 in DDD book
- Context Boundaries
  - Chapter 14 in DDD book
- DDD Community (dddcommunity.org)
  - Experience reports
  - Discussion groups

QCon

杭州站 · 2011年10月20日~22日

www.qconhangzhou.com（6月启动）

QCon北京站官方网站和资料下载

www.qconbeijing.com

全球企业开发大会

THE ANNUAL INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE