Domain-Driven Design

An Introduction

Domain-Driven DESIGN

Tackling Complexity in the Heart of Software

Eric Evans

Foreword by Martin Fowler

brief history and context

Why Bother with Domain Modeling?

focus modeling on complexity

software is not the complexity

critical complexity is
the domain itself

or at least it should be

there are exceptions

yours is probably not an exception

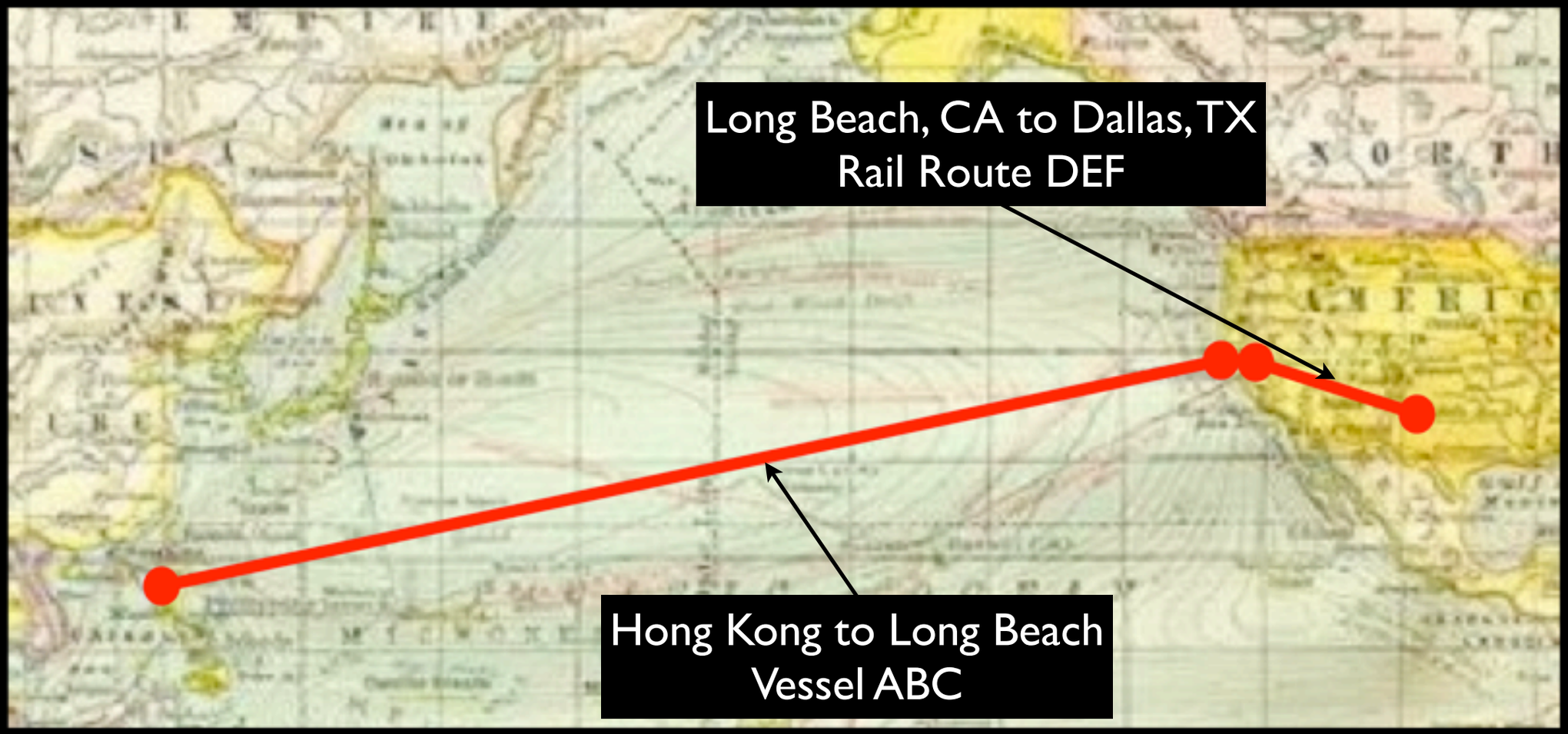
Domain

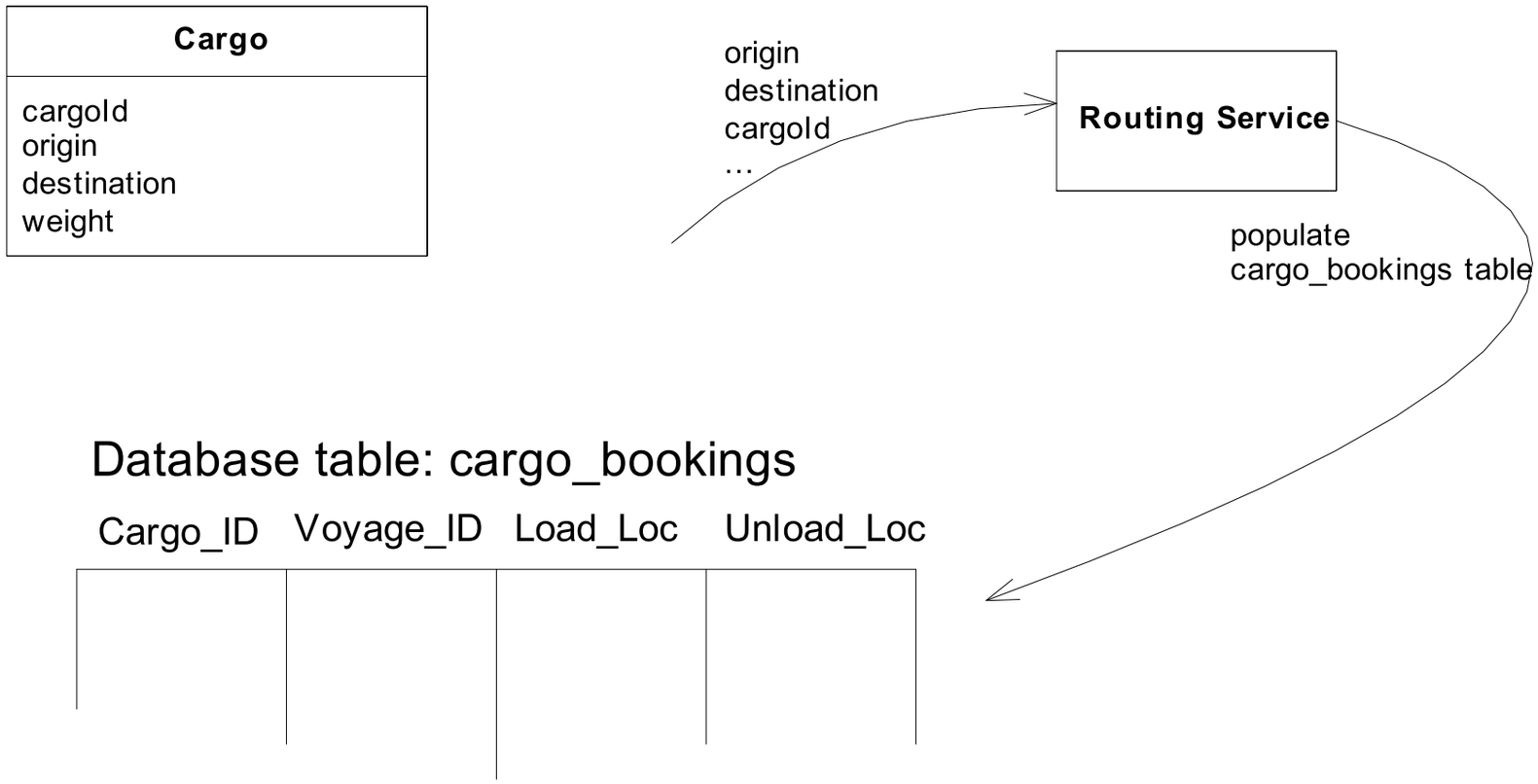
sphere of knowlege,
influence, or activity

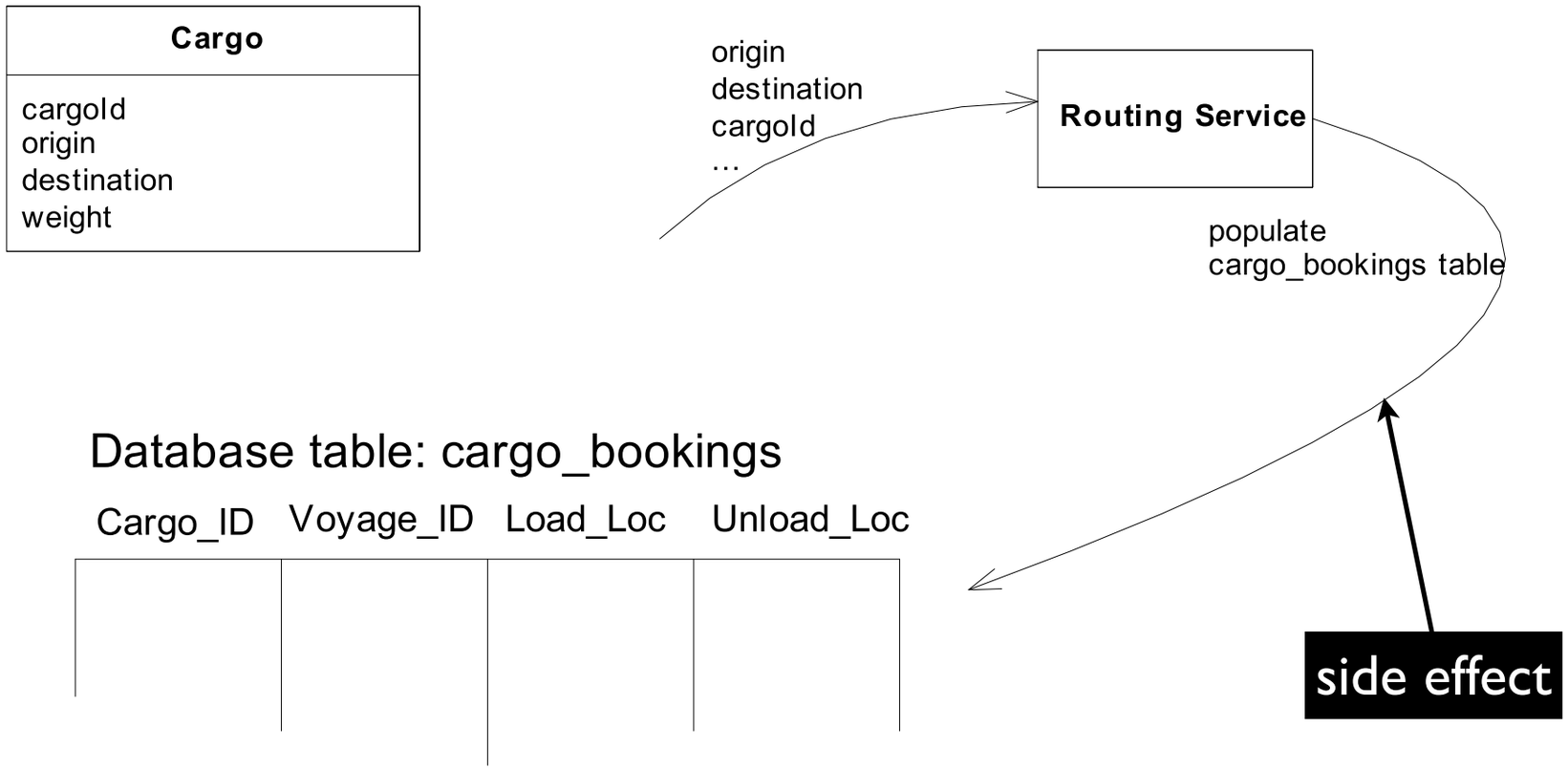
subject to which a program applies

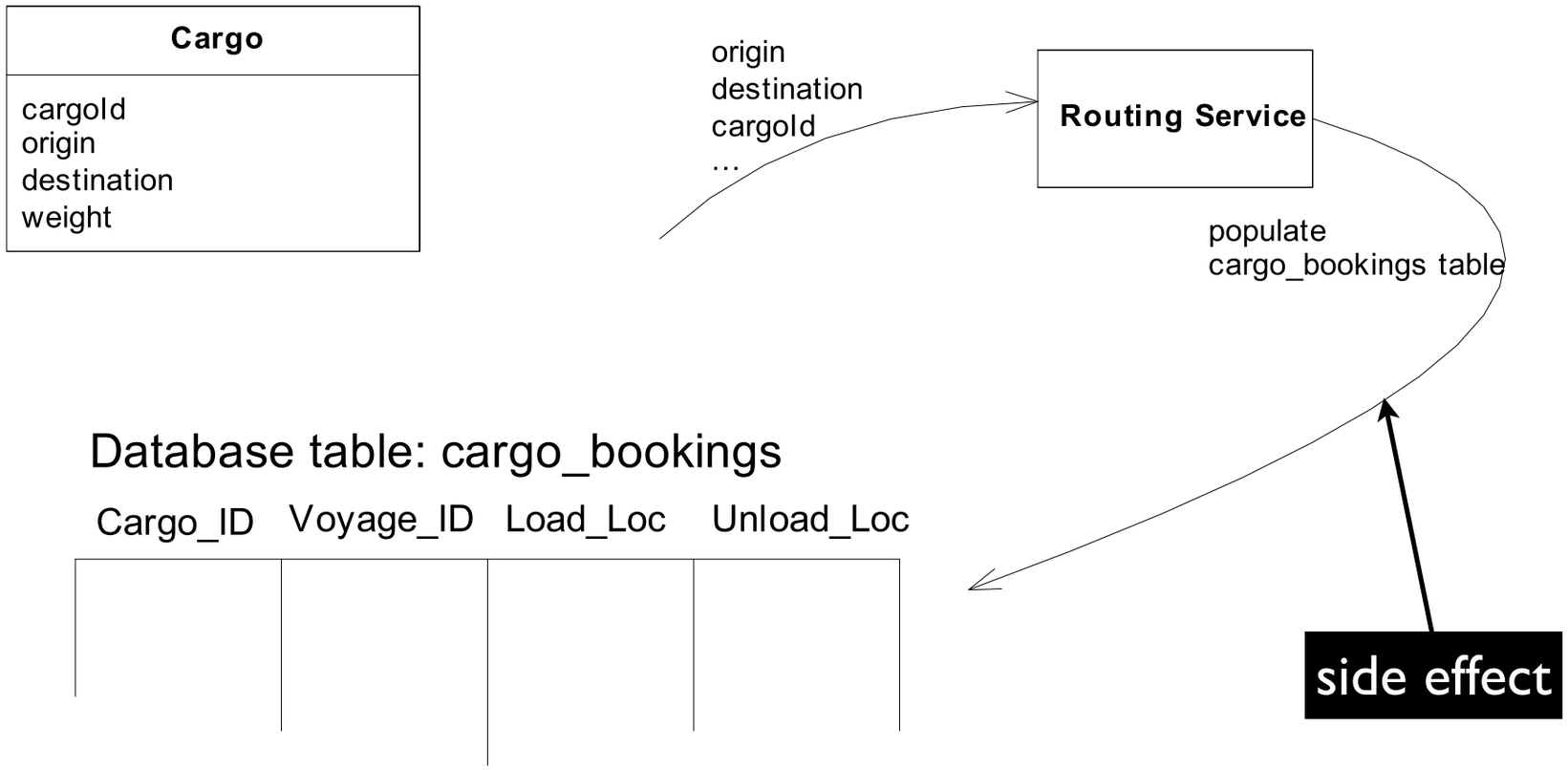
Monday, March 8, 2010


MAERSK

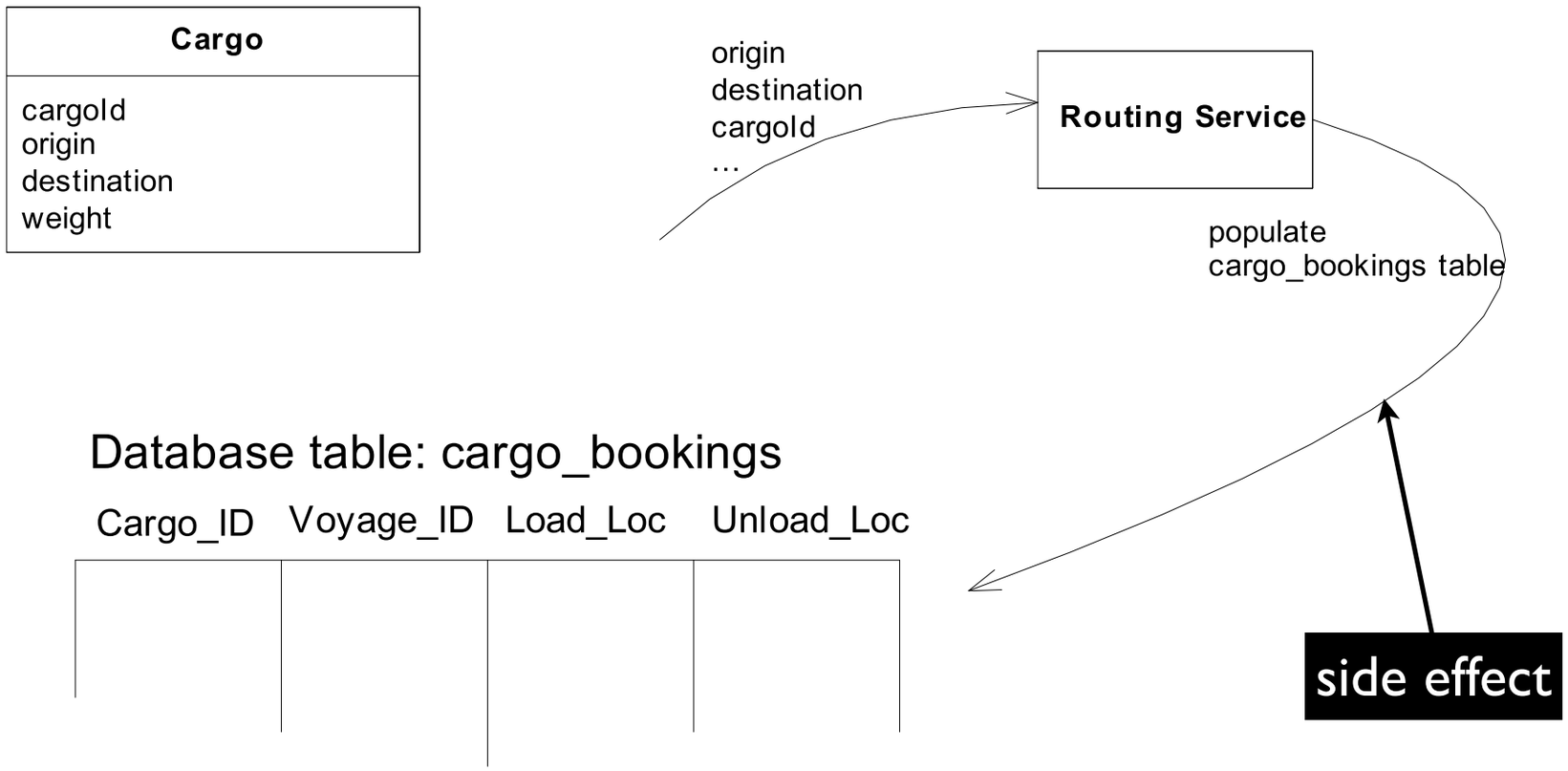

Long Beach, CA to Dallas, TX
Rail Route DEF
Hong Kong to Long Beach
Vessel ABC


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



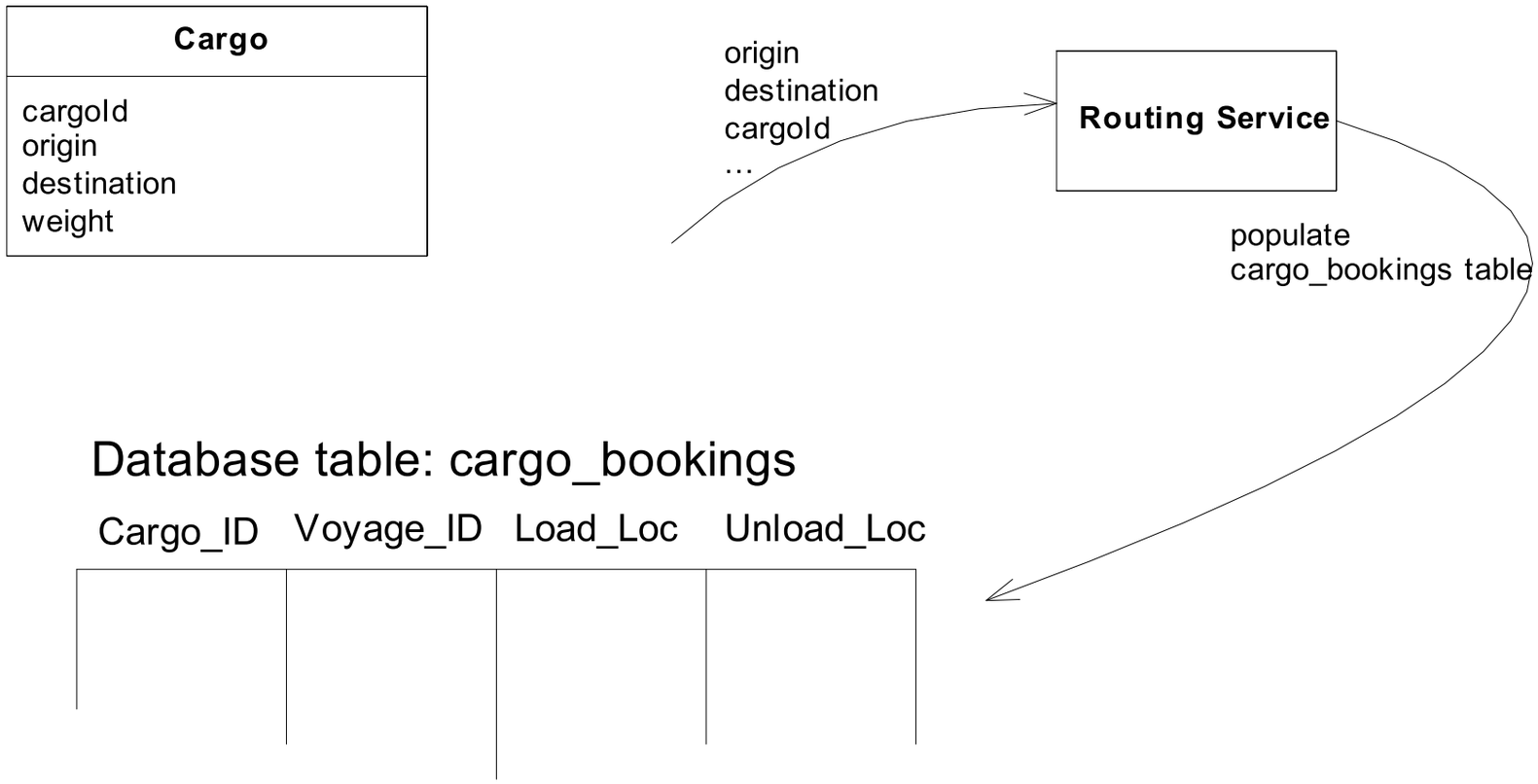
"each table row shows load and unload"



"each table row shows load and unload"

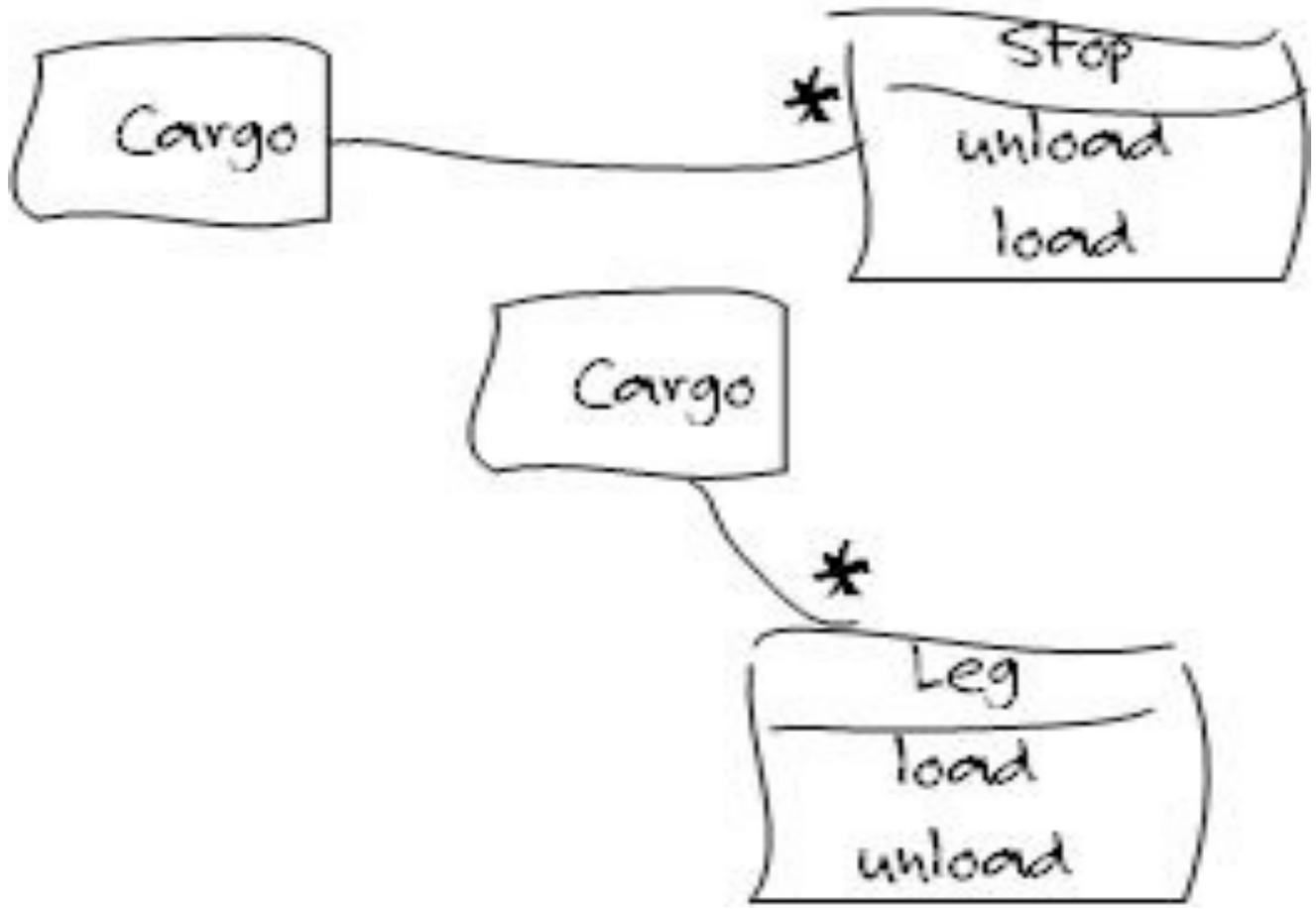
"give the Routing Service the required origin, destination, and arrival time, etc., and it will look up the stops the cargo will have to make and put them into the database"

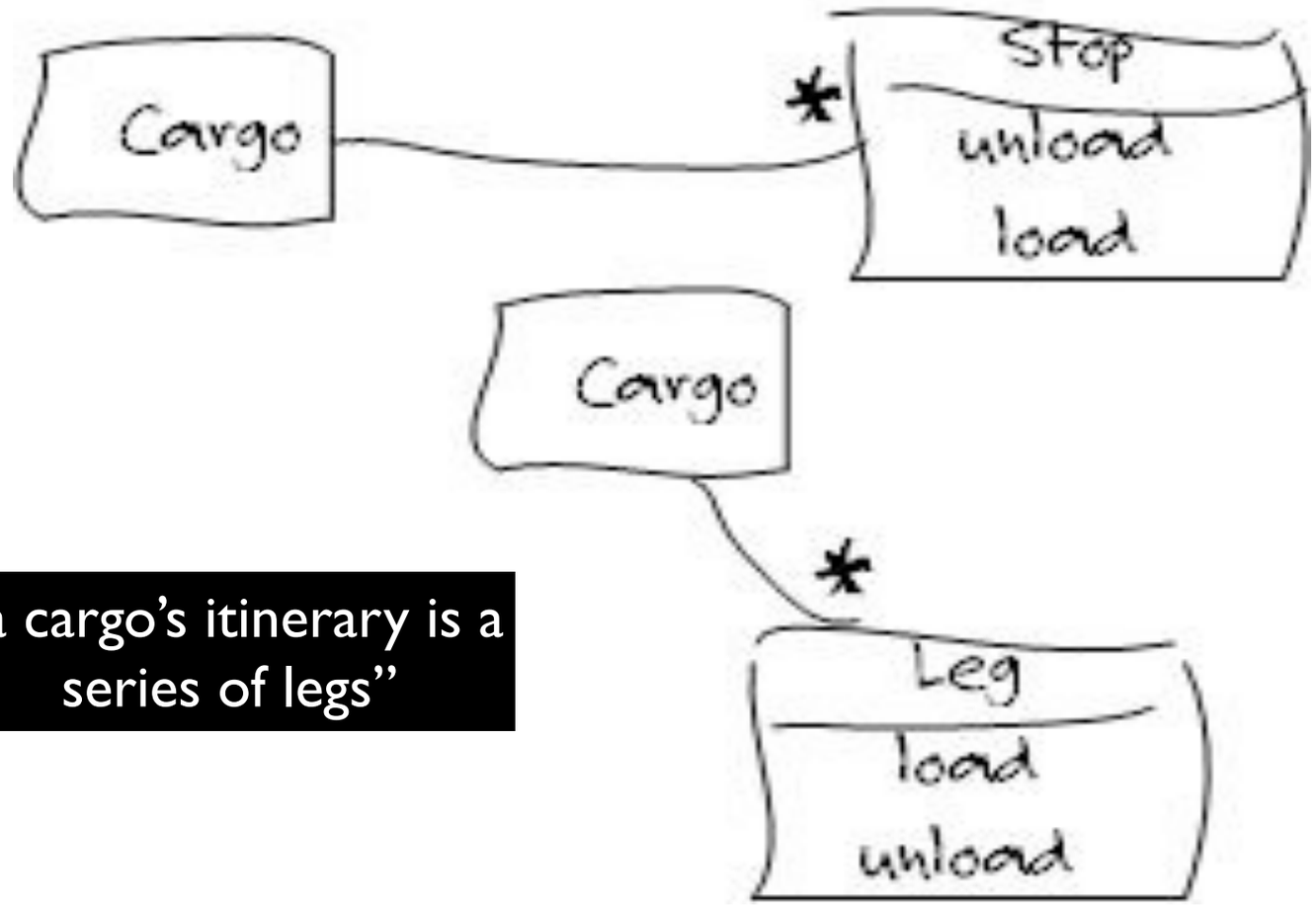
what concepts are missing?


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

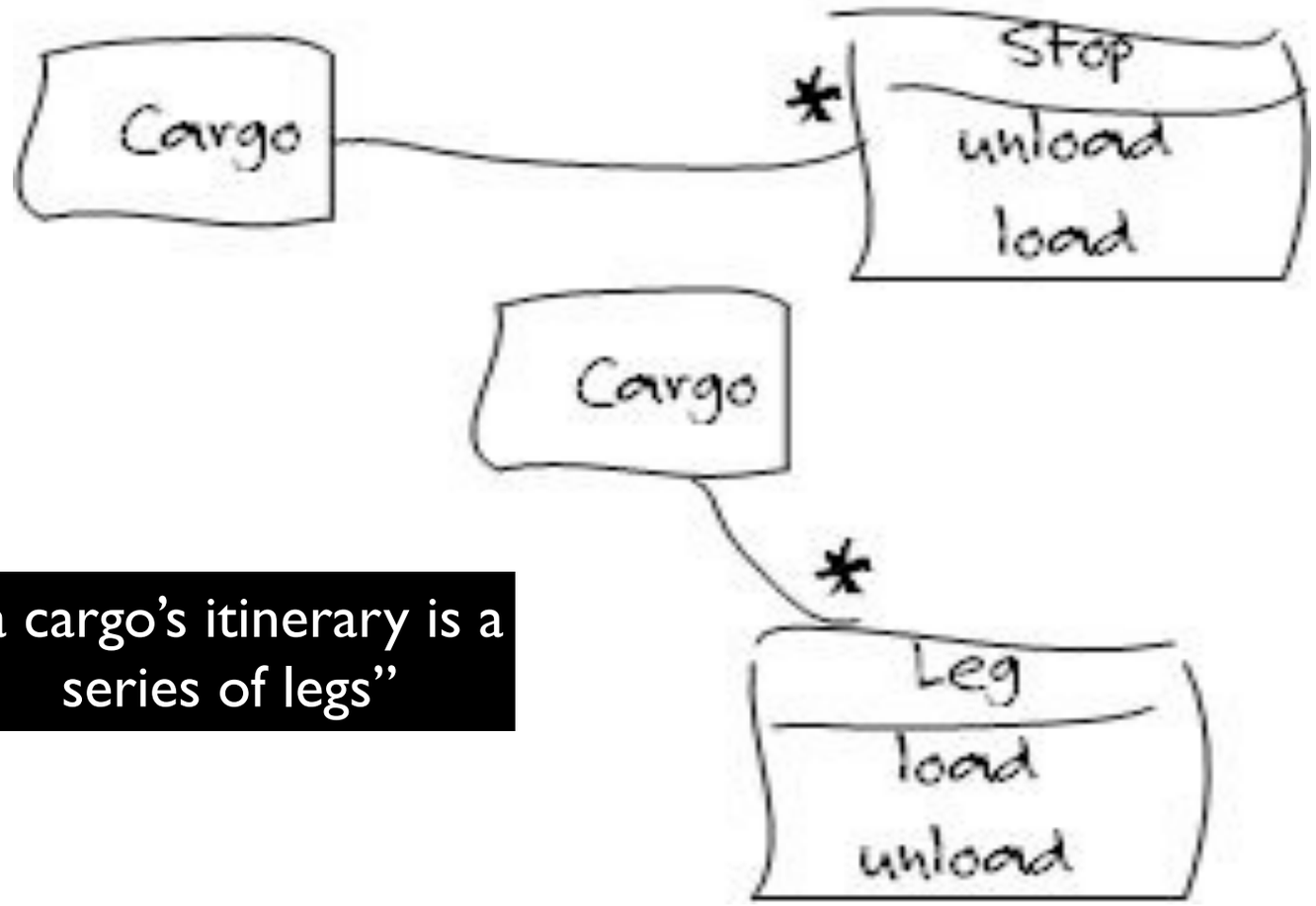

Cargo
*
Stop
unload
load

"a cargo makes a series of stops; at each stop it is unloaded from a transport and loaded onto another"


Cargo
*
Stop
unload
load
Cargo
*
Leg
load
unload

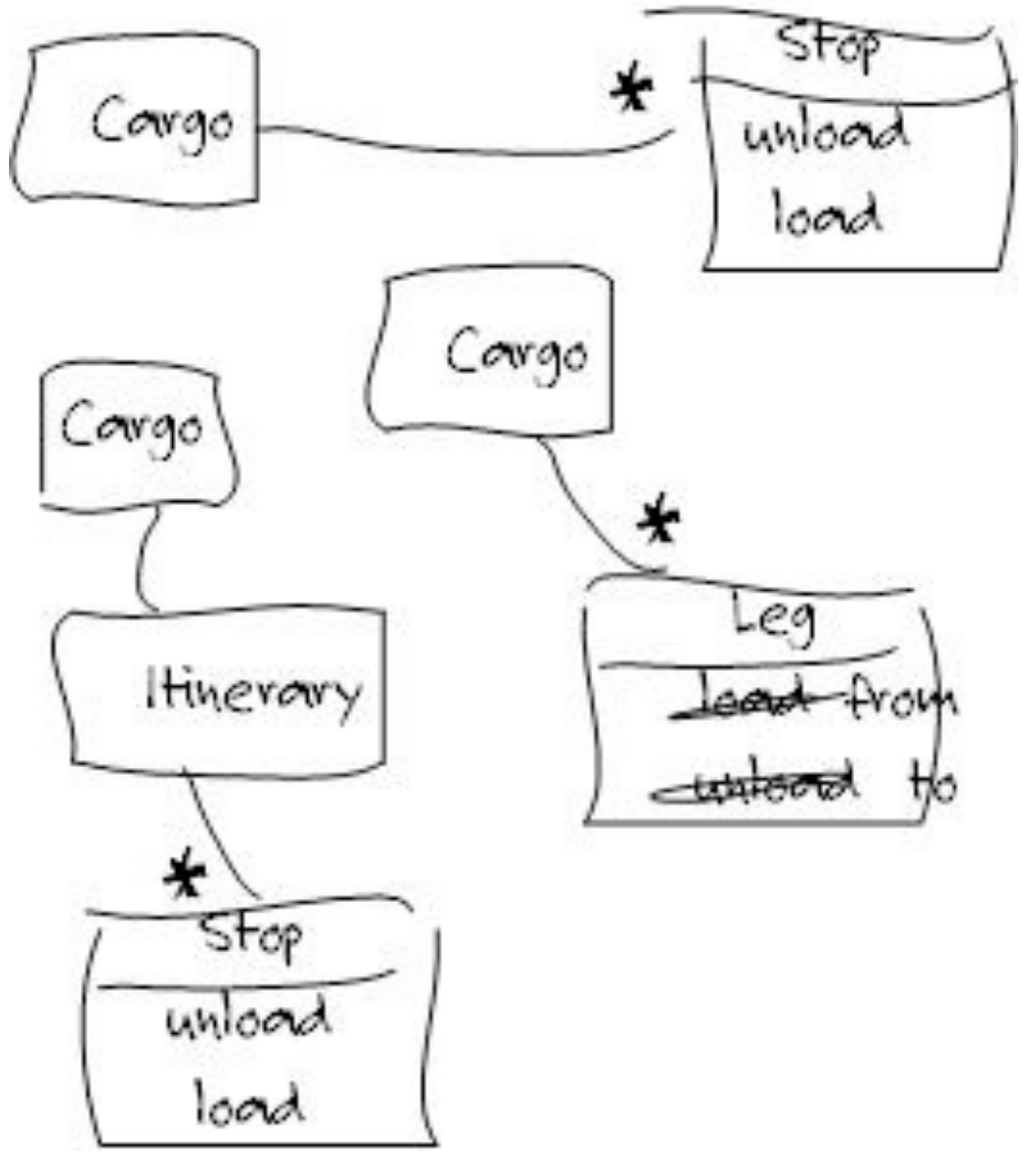


"a cargo's itinerary is a series of legs"

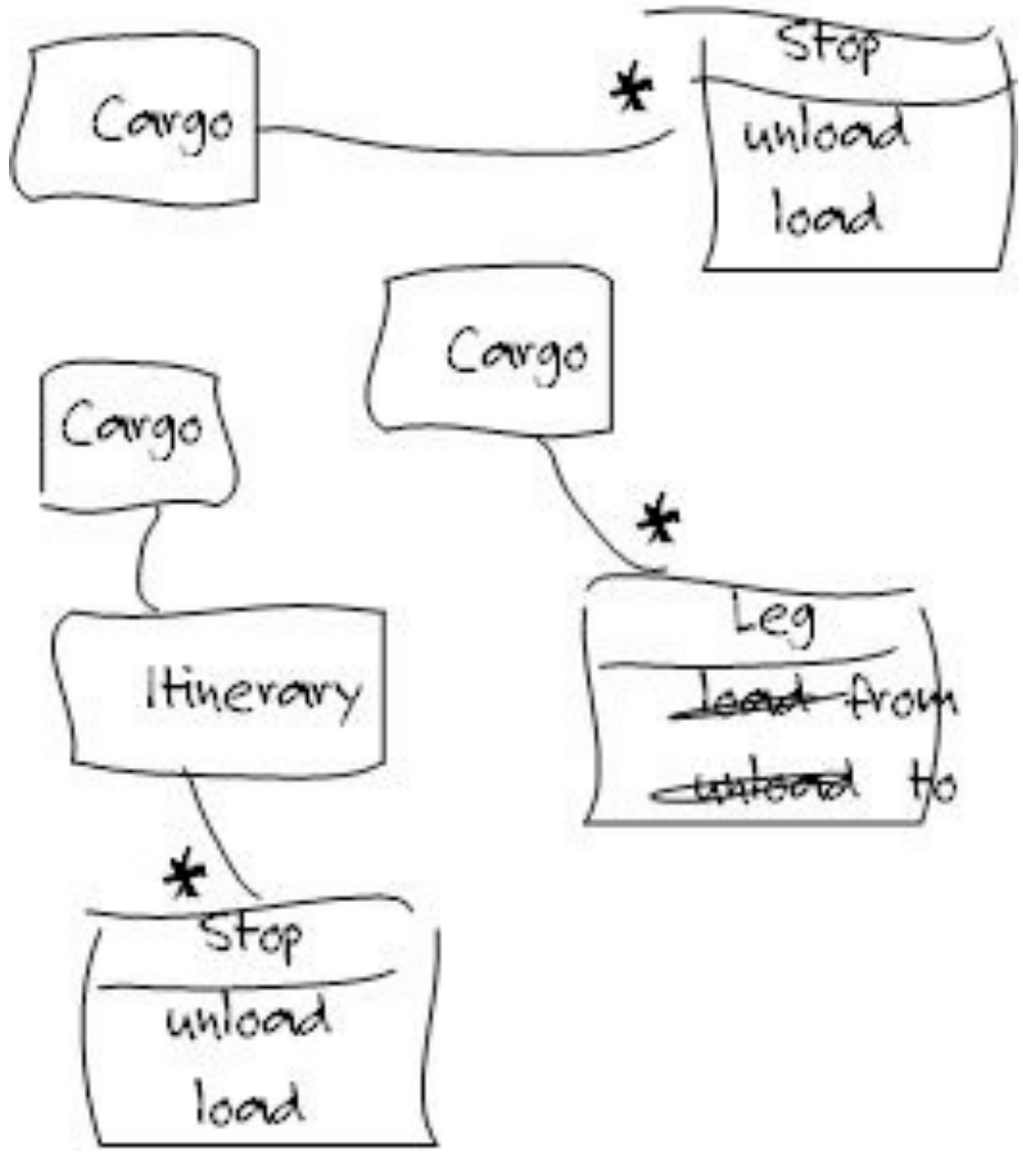


"a cargo's itinerary is a series of legs"

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


Cargo
*
Stop
unload
load
Cargo
Cargo
*
Leg
~~load~~ from
~~unload~~ to
Itinerary
*
Stop
unload
load

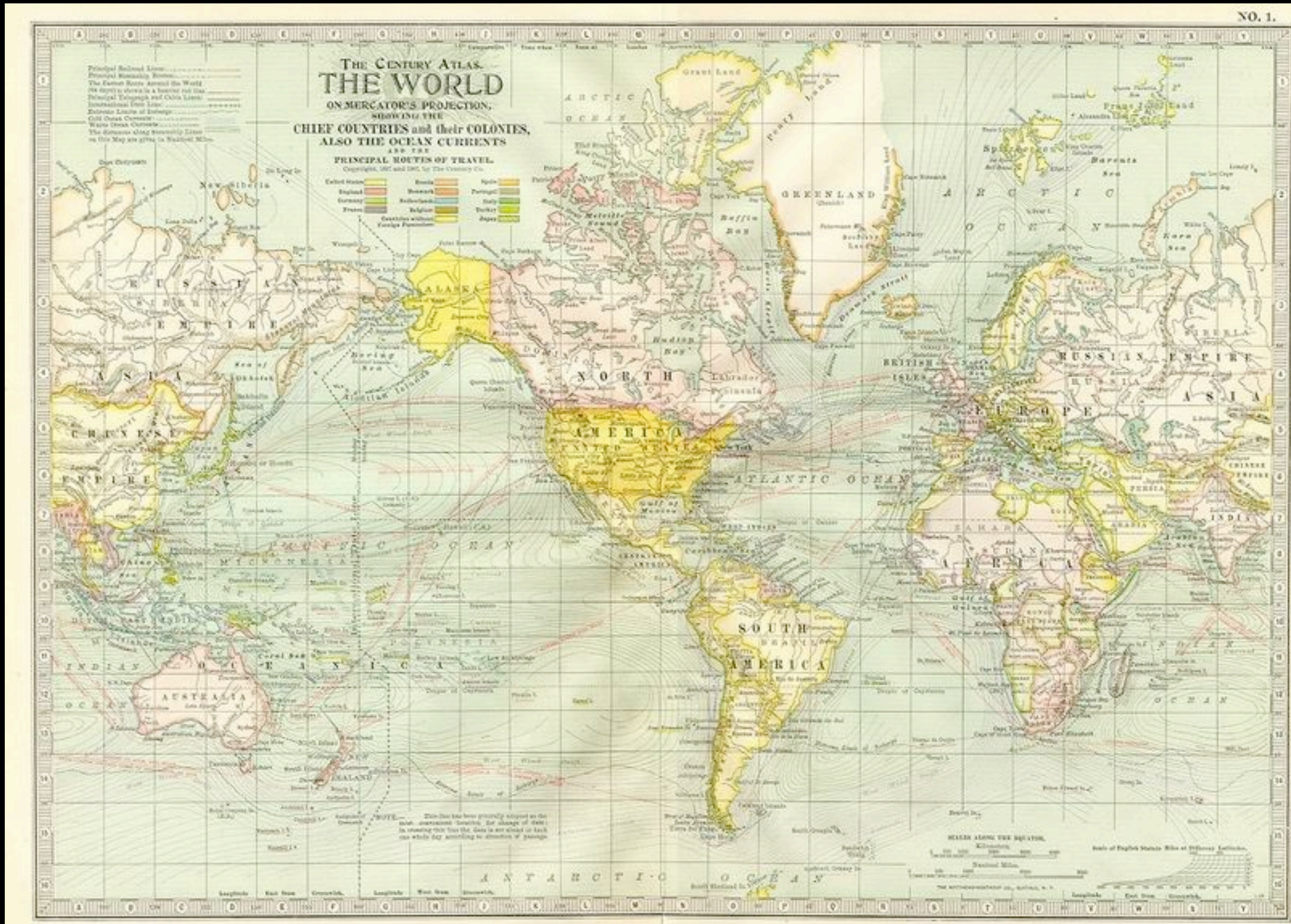
"a cargo's itinerary gives instructions for each stop, to unload and load onto a different stop"



stops or legs?
which is better?

What Is A Model?

Monday, March 8, 2010



Monday, March 8, 2010

Model

system of absractions

describes selected

aspects of a domain

used to solve problems
in domain

Domain Model

not a diagram, idea
diagram conveys

a selective abstraction
of expert knowledge

a model serves a particular use

not “as realistic as possible”

allows software to

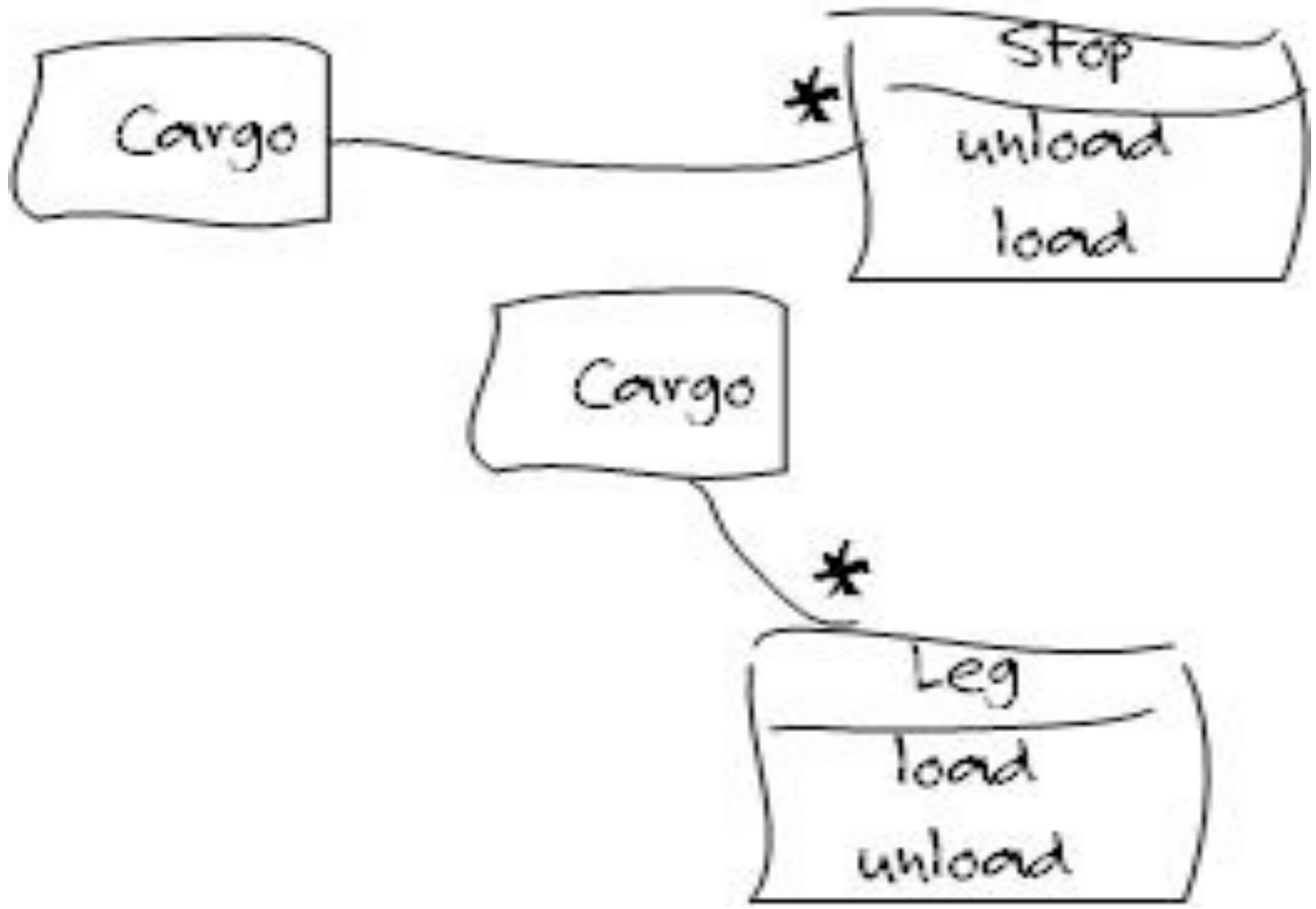
enter the domain

models inexpressible in code are irrelevant

stops or legs?
which is more useful?

stops or legs?
which is more useful?

useful for *what*?


Cargo
*
Stop
unload
load
Cargo
*
Leg
load
unload

when modeling, choose concrete reference scenarios

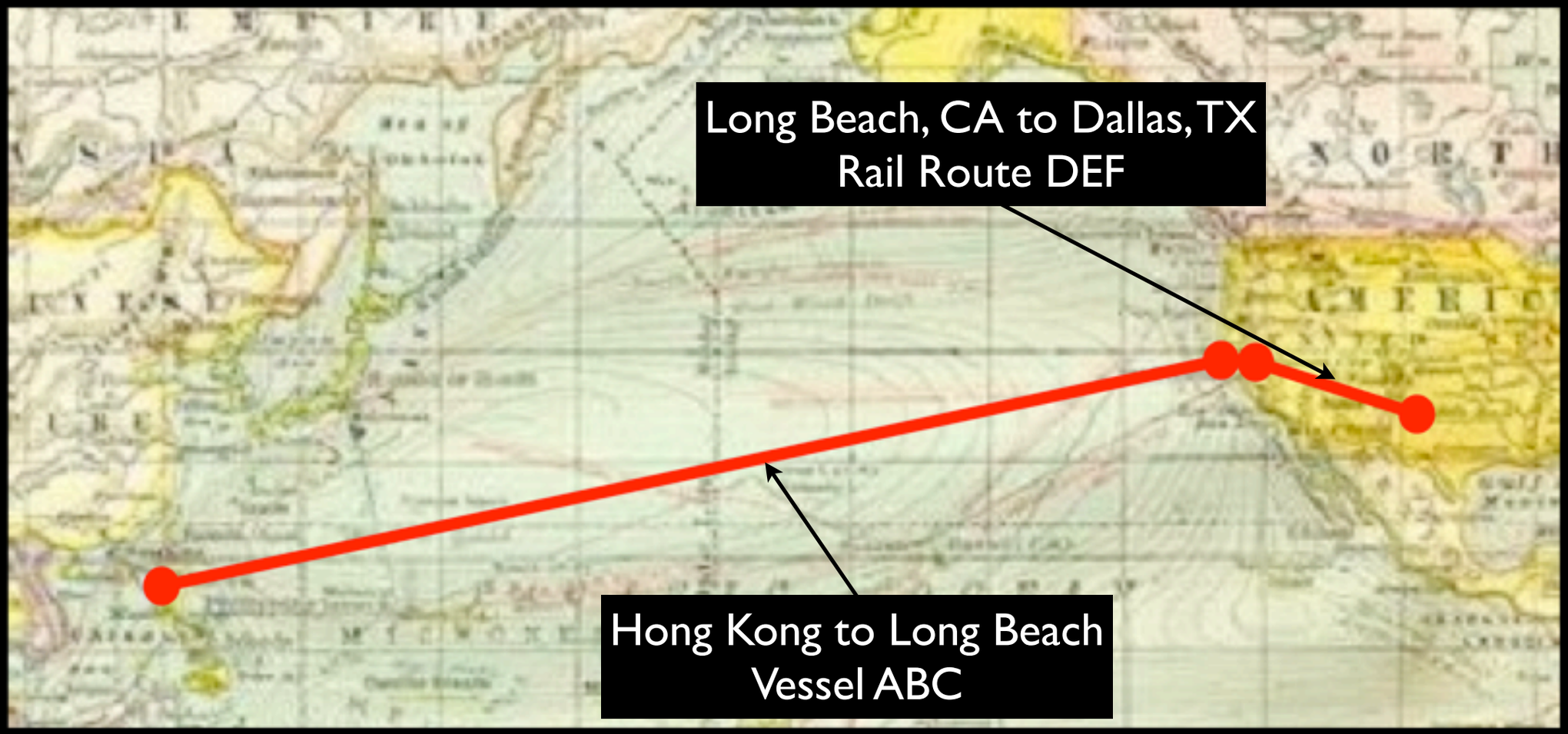
shipping reference scenarios

shipping reference scenarios

As a customer, route and book a new shipment

shipping reference scenarios

As a customer, route and book a new shipment


Long Beach, CA to Dallas, TX
Rail Route DEF
Hong Kong to Long Beach
Vessel ABC

shipping reference scenarios

shipping reference scenarios

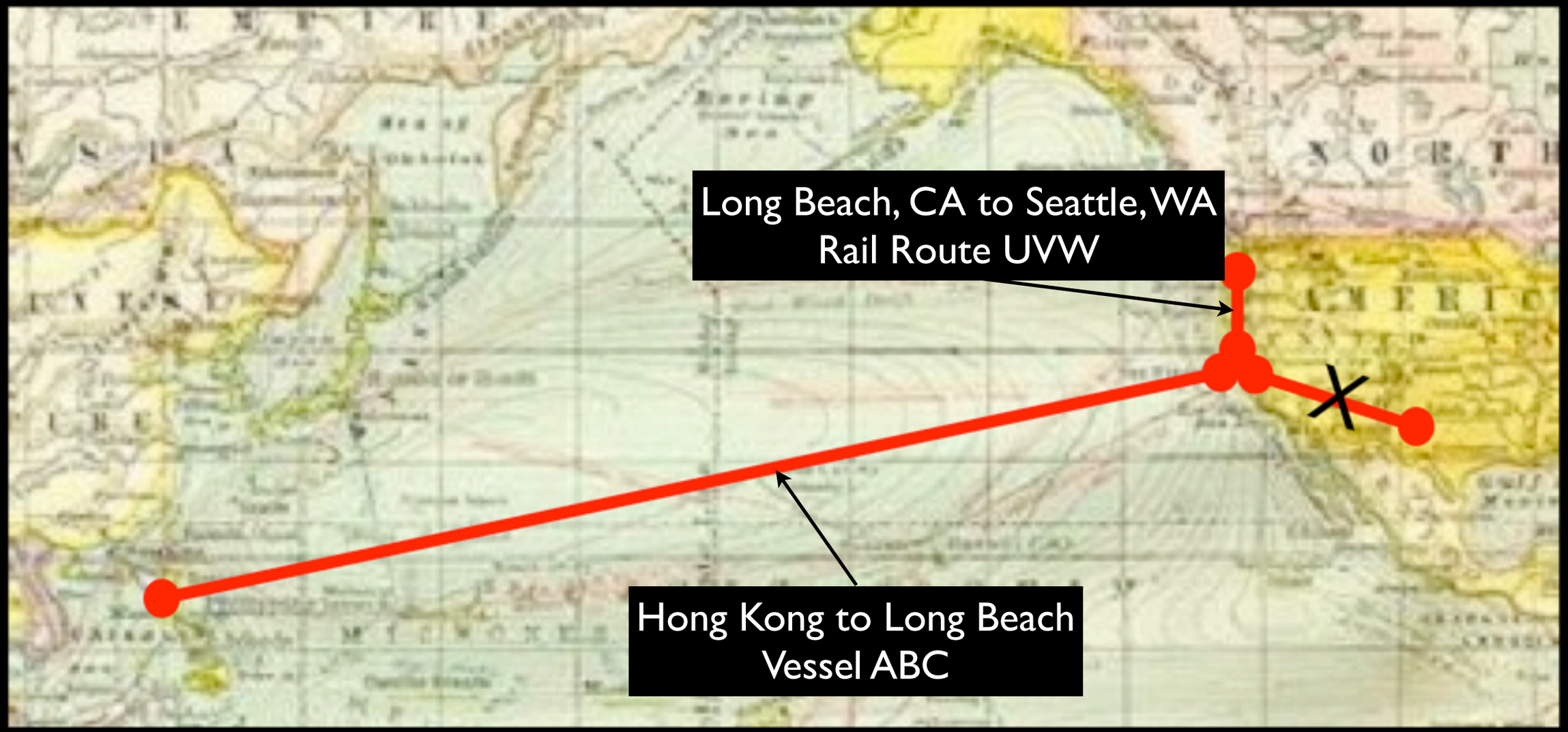
As a customer, route and book a new shipment

shipping reference scenarios

As a customer, route and book a new shipment

As a customer, I can reroute my shipment mid-transit


Hong Kong to Seattle, WA
Vessel XYZ


Long Beach, CA to Seattle, WA
Rail Route UVW
Hong Kong to Long Beach
Vessel ABC

stop-based model

pseudo-code:

find next Stop.

remove loading portion of next Stop, and all subsequent Stops.

ask RoutingService for itinerary from next Stop location to new destination.

insert loading portion of first Stop into remainder of next Stop

append remaining Stops of new Itinerary to truncated Itinerary

leg-based model

pseudo-code:

find current Leg.

truncate Itinerary after current Leg.

ask RoutingService for itinerary from current Leg end to new destination.

append all Legs of new Itinerary to truncated Itinerary

shipping reference scenarios

shipping reference scenarios

As a customer, route and book a new shipment

shipping reference scenarios

As a customer, route and book a new shipment

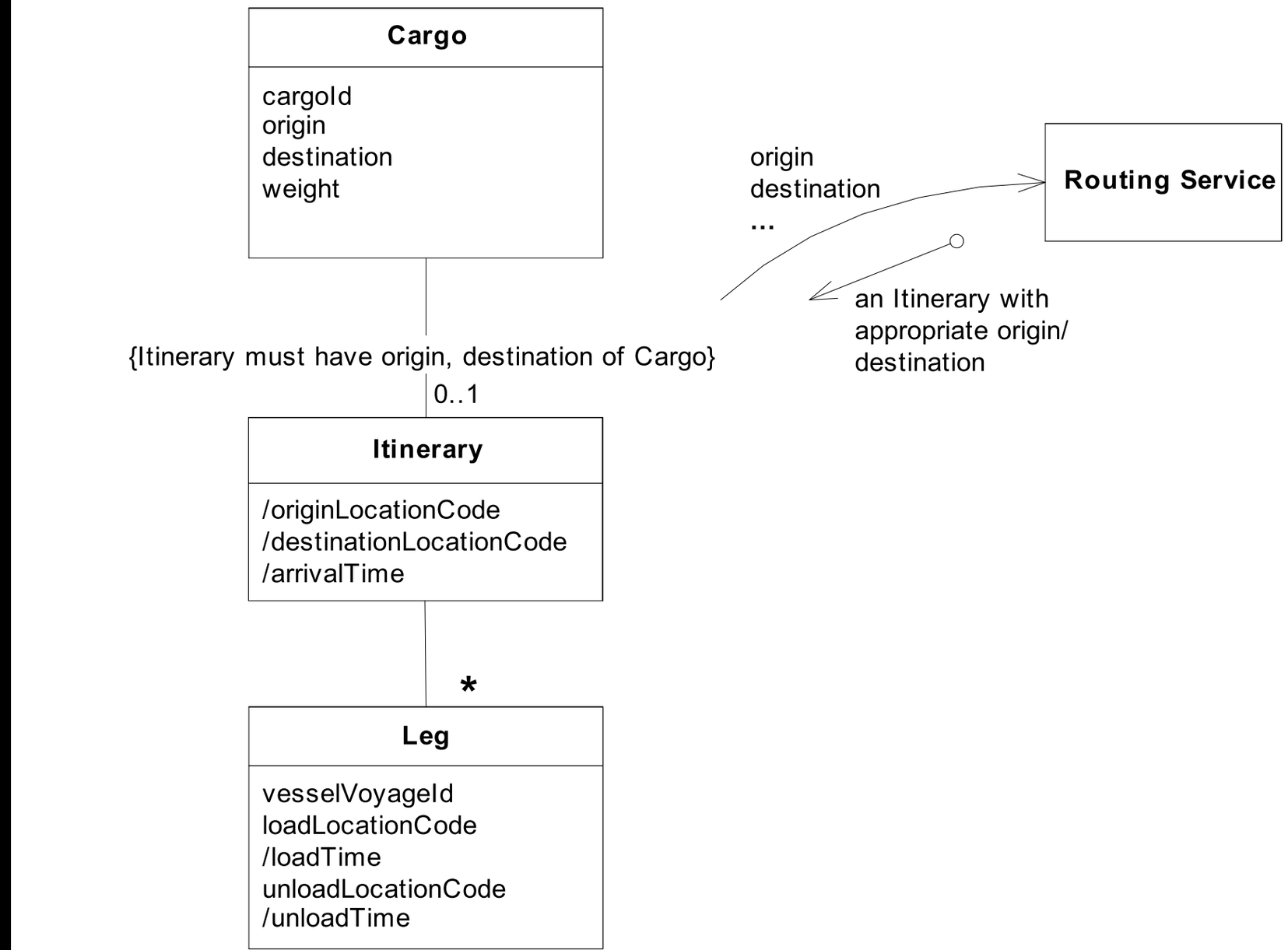
As a customer, I can reroute my shipment mid-transit

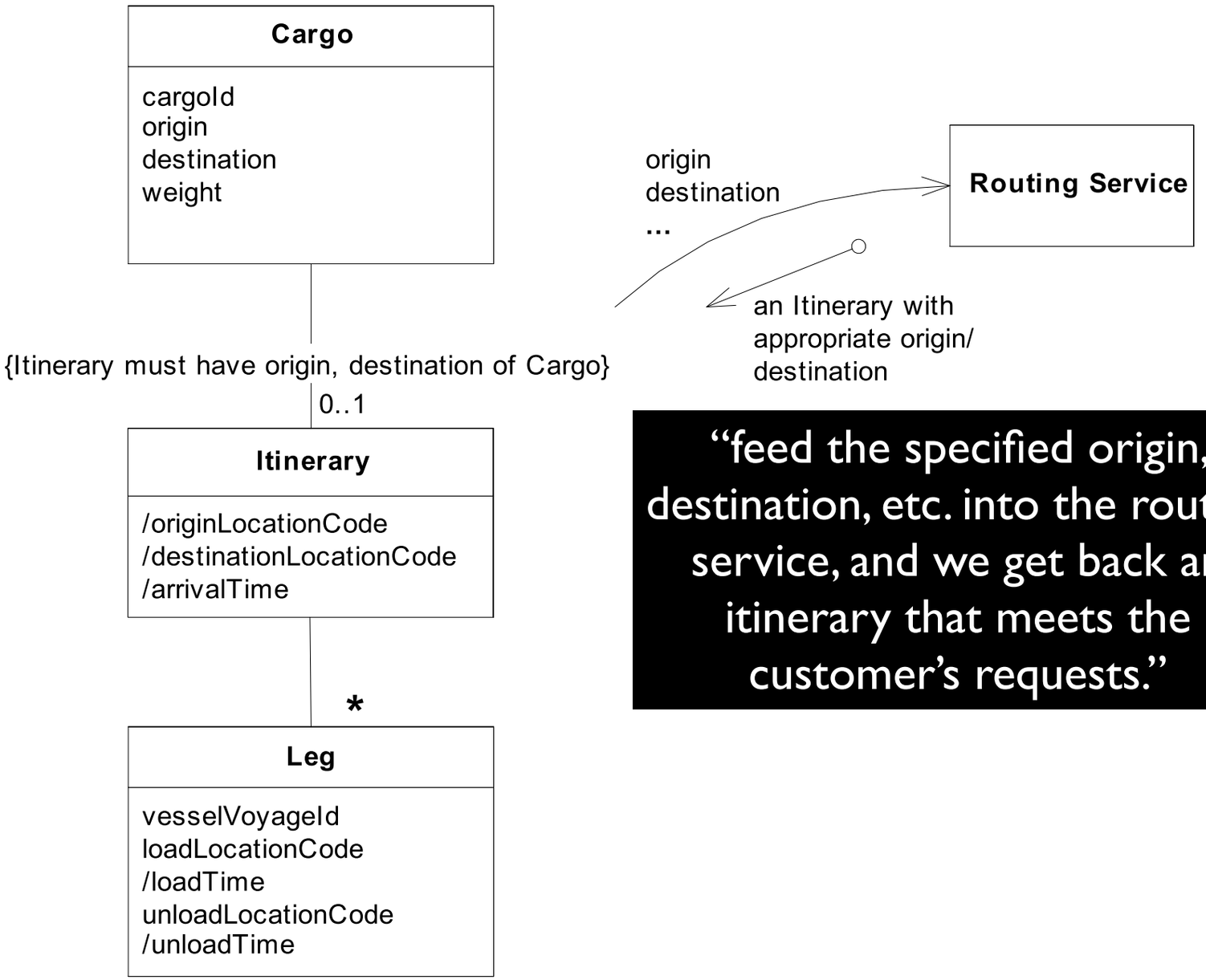
shipping reference scenarios

As a customer, route and book a new shipment

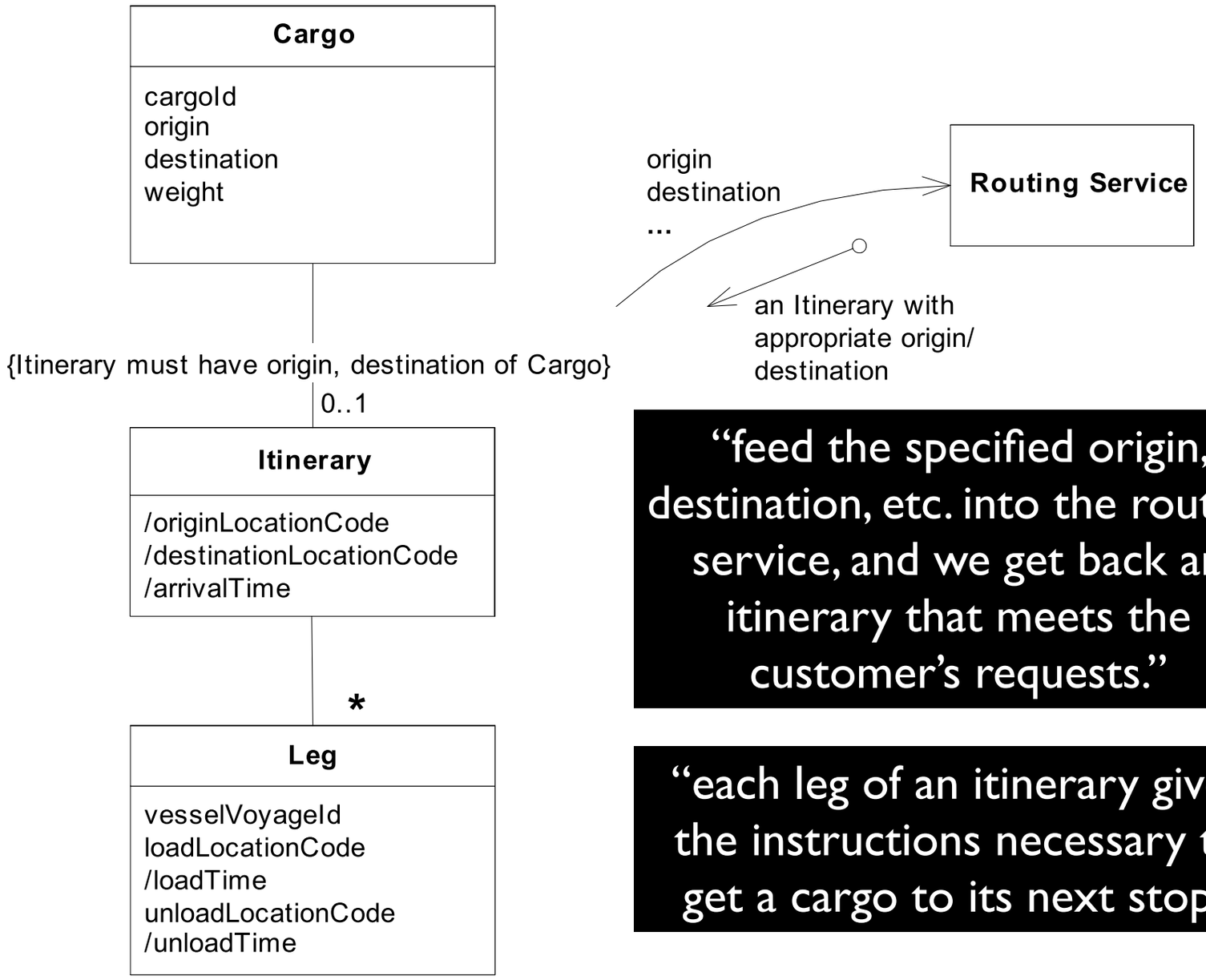
As a customer, I can reroute my shipment mid-transit

As a freight terminal operator, I can issue purchase orders to other vendors in the the port before each voyage arrival for unloading, loading, storage, and routing within the terminal


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, etc. into the routing service, and we get back an itinerary that meets the customer's requests."



"feed the specified origin, destination, etc. into the routing service, and we get back an itinerary that meets the customer's requests."

"each leg of an itinerary gives the instructions necessary to get a cargo to its next stop"

Multiple Models

context determines a model's usefulness

some models cause more implementation challenges

separating model and implementation causes irrelevance

Ubiquitous Language

language structured around the domain model

used by all team members

connects team activity to the software

Context

setting that determines meaning of word

Key DDD Concepts

experimentation

interaction with domain

experts

ubiquitous language

context

parting thought on Agile and DDD

all things computed

Barry Hawkins

http://alltc.com

coaching/mentoring

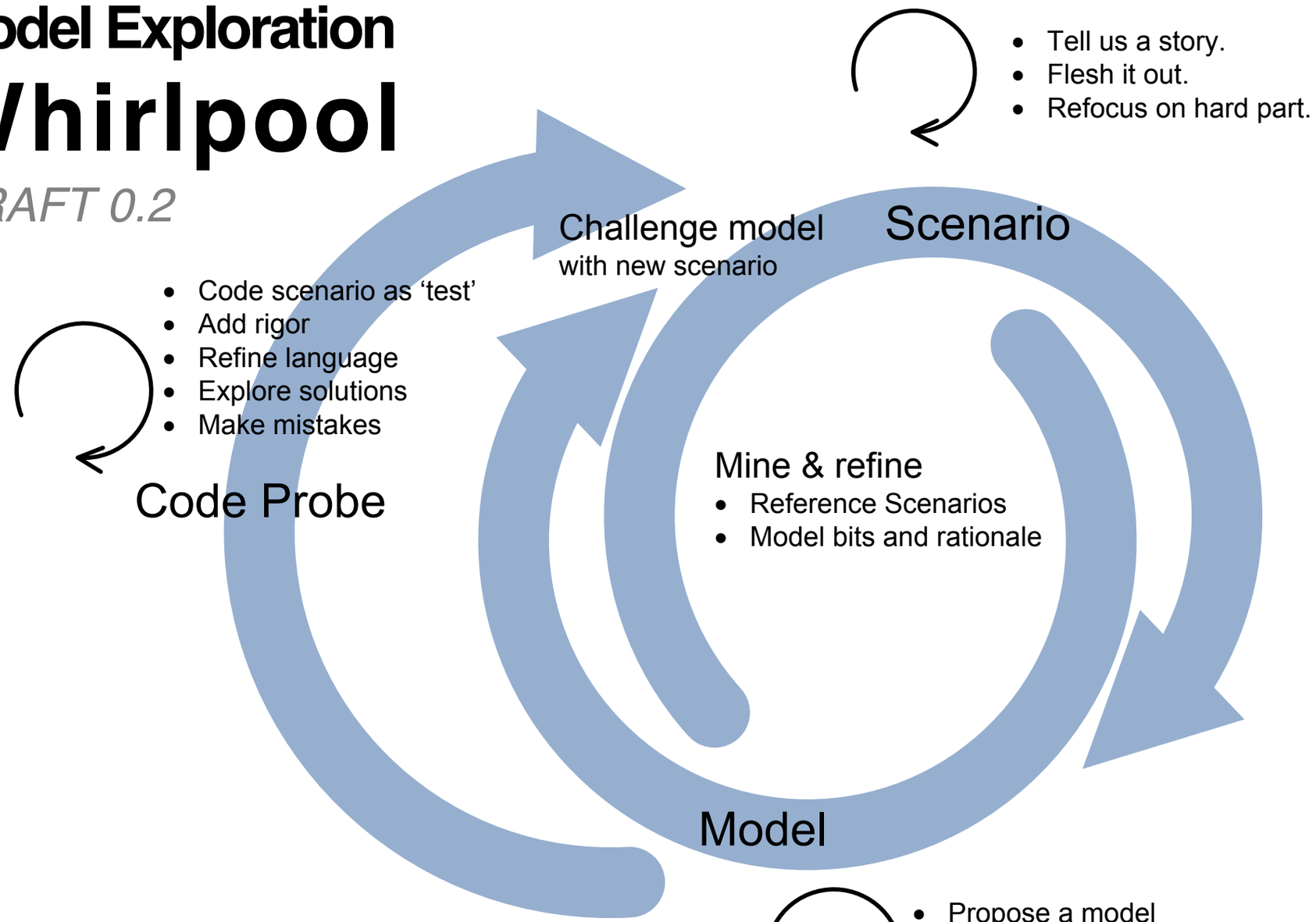
agile software development

domain-driven design

Model Exploration

Whirlpool

DRAFT 0.2



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

- 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/processdraft