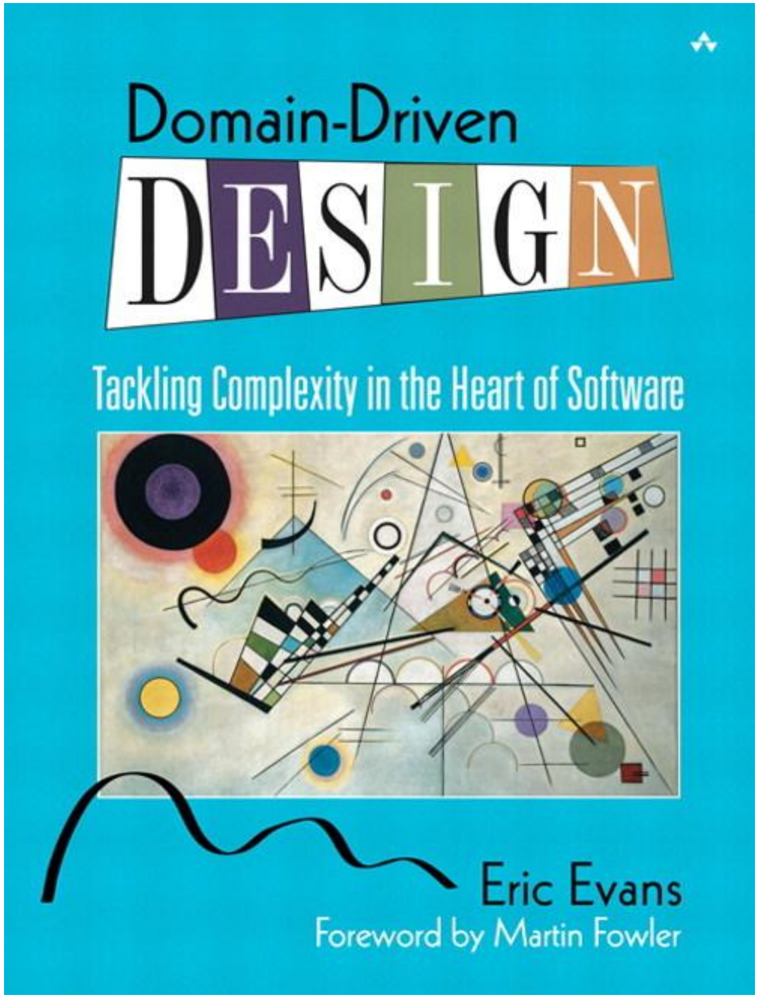
Domain-Driven Design

CS 618

Feb 28, 2012

Bill Kidwell


Domain-Driven
DESIGN
Tackling Complexity in the Heart of Software
Eric Evans
Foreword by Martin Fowler

Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

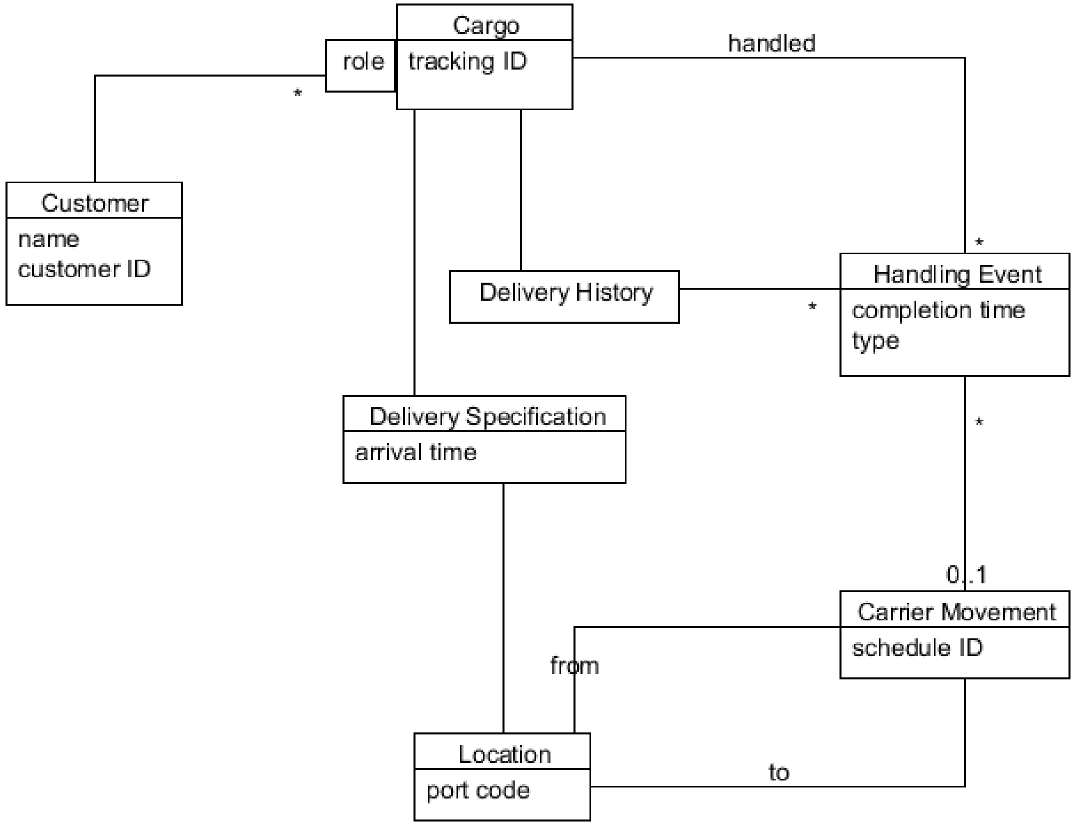
Cargo Shipping System

- Chapter 7: Using the Language: An Extended Example
- Initial Requirements:
  - Track key handling of customer cargo
  - Book cargo in advance
  - Send invoices to customers automatically when the cargo reaches some point in its handling

Starting Sample Model

- In the real world, it would take some time to get to this point
- Evans covers the discovery process in Part III
- This section covers fine-tuning the domain model

Figure 7.1



Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

Three User-Level App Functions

- *Tracking Query* that can access past and present handling of a particular Cargo
- *Booking Application* that allows a new Cargo to be registered
- *Incident Logging Application* that can record each handling of the Cargo

Domain Objects

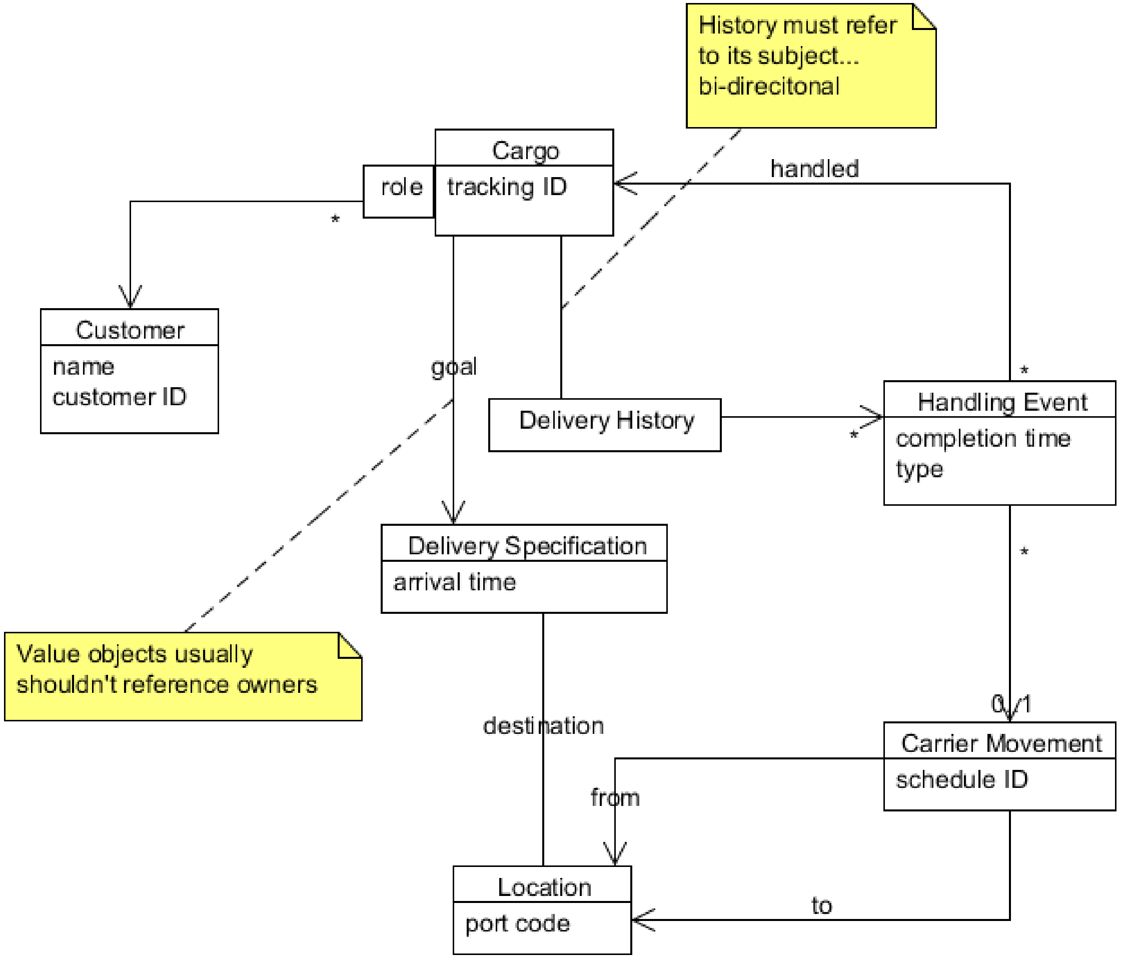
- Customer – Entity object
  - Internally assigned ID
- Cargo – Entity Object
  - Tracking ID
- Carrier Movement – Entity Object
  - Identified by code on the shipping schedule
- Handling Event
  - Identified by Cargo ID, Completion time, type

Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

Domain Objects

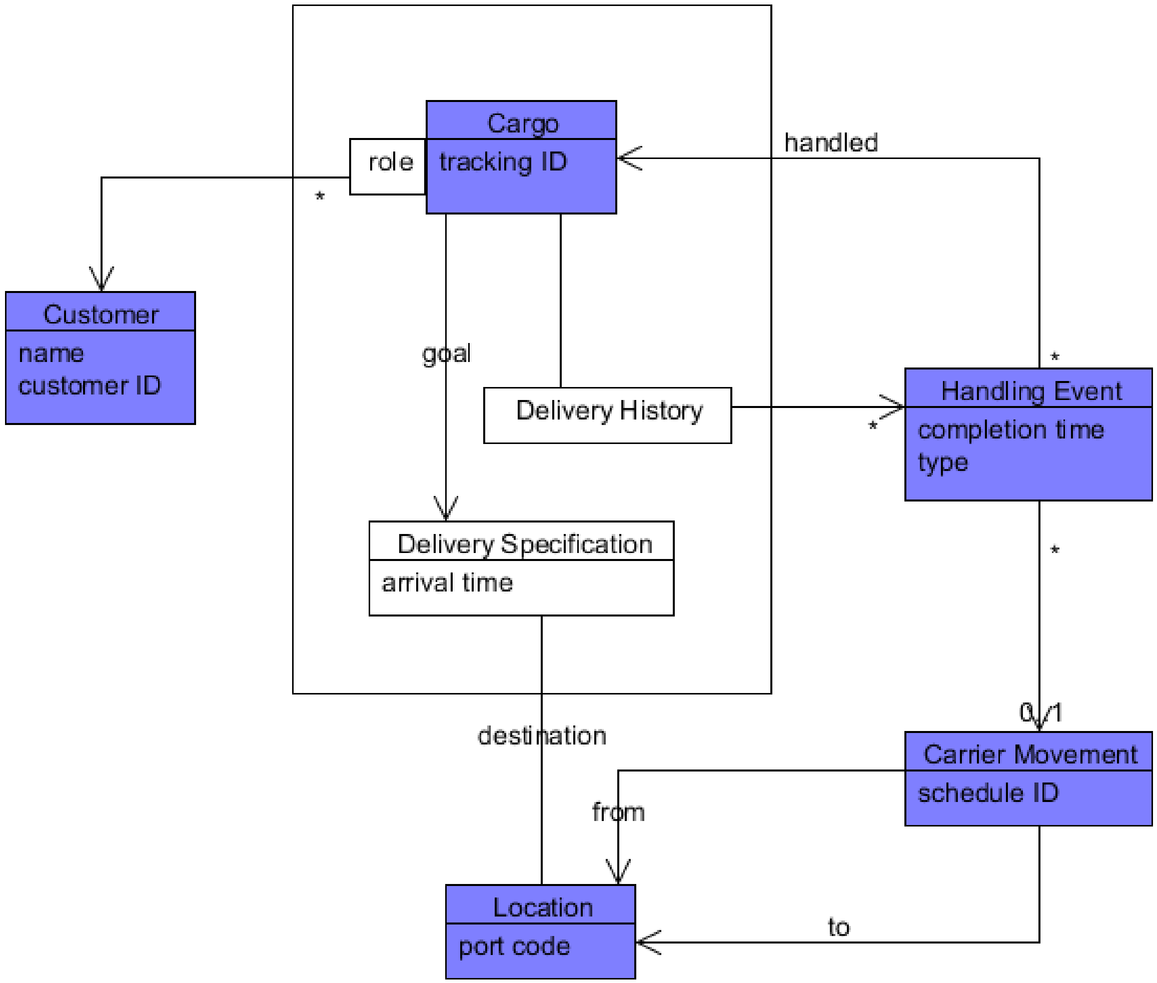
- Location
  - Probably part of an existing geographical model
- Delivery History
  - Not interchangeable, so they are entities
  - One-to-one relationship with Cargo, its identifier
- Delivery Specification – Value Object
  - Hypothetical Delivery History state
- Role – Value Object

Associations



Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

Aggregate Boundaries



Domain-Driven Design: Tackling Complexity in the Heart of Software by Eric Evans

Selecting Repositories

- Go back to the requirements to see what repositories are needed…
- In order to take a booking, we must select the customer and a Location
  - Customer Repository
  - Location Repository
- Activity Logging Application needs to allow the user to look at the Carrier Movement that a Cargo is being loaded on, and which Cargo is being loaded
  - Carrier Movement Repository
  - Cargo Repository
- Currently no need for a Handling Event Repository

Refining the Model

- The author steps through scenarios
- As he does, he looks critically at the design
- Looking at the design, he points out that each time we add a handling event, we have to update the Cargo in order to update the Delivery History
    - This causes contention
        - Handling Events happen often, and have to be fast
    - They remove the state from Delivery History and replace it with a query of Handling Events
    - This causes the addition of the Handling Event Repository