

# **Learn the Business with Domain Models**

Joaquim Baptista



**FARFETCH**

## MAJOR OBJECTIVES

1. What are domain models?
2. Abstract away technical detail to learn the business.
3. Describe the context to learn the business.
4. Should you use domain models?

The forever evolving

# LARGE SYSTEM

## **Large systems for complex problems**

Interesting projects typically pursue a complex problem.

## **Multiple teams, multiple skills**

Different teams struggle with different parts of the problem, using different skills, often while training junior team members.

## **Multiple mental models**

Teams develop and evolve mental models of their own part of the problem.

## **Models grow apart**

Over time, models from different teams become subtly incompatible with each other, leading to inconsistencies in products and processes.

## **Forever incomplete, imperfect**

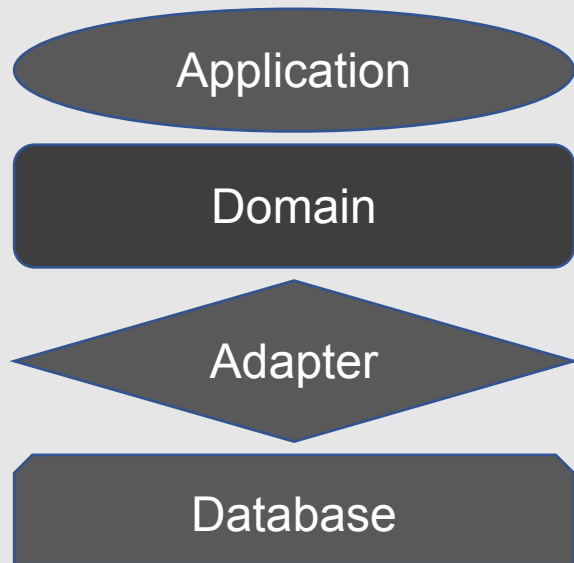
Large systems keep evolving, forever pursuing an evolving business reality in an imperfect way.

Eric Evans, 2003

Domain-Driven Design:  
Tackling Complexity  
in the Heart of Software.

Addison-Wesley.

DDD Lisbon Meetup, 16-Mar-2018



## LAYERED SOFTWARE

A good domain layer:

- Is understandable both by business experts and software experts.
- Supports the flexible (supple) evolution of software.

## Eric's Domain Models

In Eric's book, the domain model is the:

1. Software layer with objects.
2. Diagrams or documents with definitions and interactions.
3. Ideas inside people's heads.

```
class Square {  
    ...  
}  
class Triangle {  
    ...  
}  
class Circle {  
    ...  
}
```

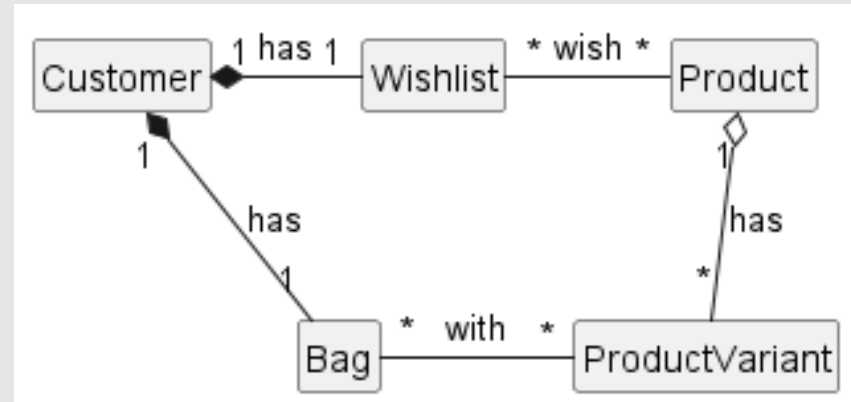


## Eric's Domain Models

Eric typically uses UML class diagrams, or explanatory diagrams in any format.

Diagrams are complemented with conversation or with written text.

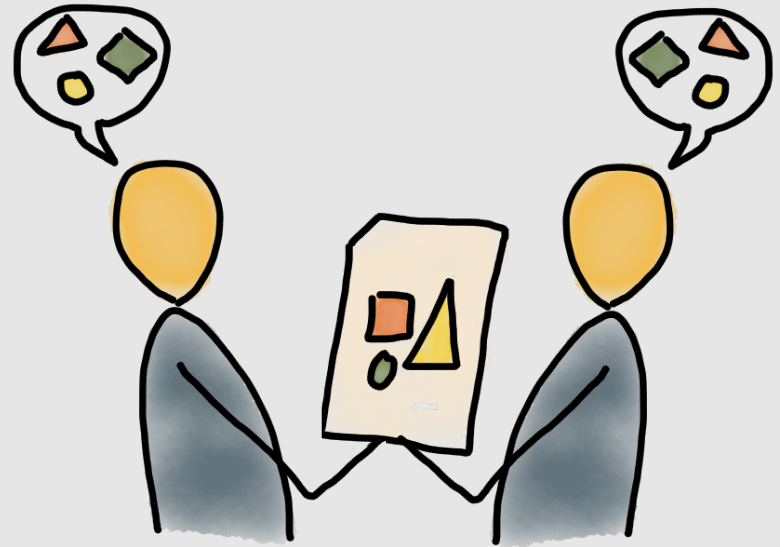
Models support brainstorming and evolve continuously through collaboration between business experts and software experts.



## Domain Models for Communication

Domain models are useful even if the software does not have a clearly defined domain layer.

- Capture knowledge of domain.
- Teach domain to newcomers.
- Align different people.
- Define common concepts and language.
- Enable the flexible evolution of ideas.





All models are wrong,  
but some are useful.

– George Box, 1976

```
class Square {  
  ...  
}  
class Triangle {  
  ...  
}  
class Circle {  
  ...  
}
```

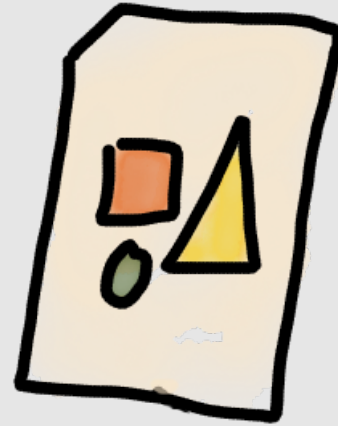


## Domain Models for Technical Writers

What is a good domain model for writers?

- Captures knowledge at a level that bridges business with software.
- Defines concepts and language.
- Provides insight on business purposes.
- Justifies decisions.

Domain models are a systematic way to gather knowledge.



**Abstract away  
technical detail  
to learn the business**

## Technical Writing Mission at Farfetch (2018-)

1. Document public APIs.
2. Also explain the business  
that can be done  
through the public APIs.
3. Also help to define the business.

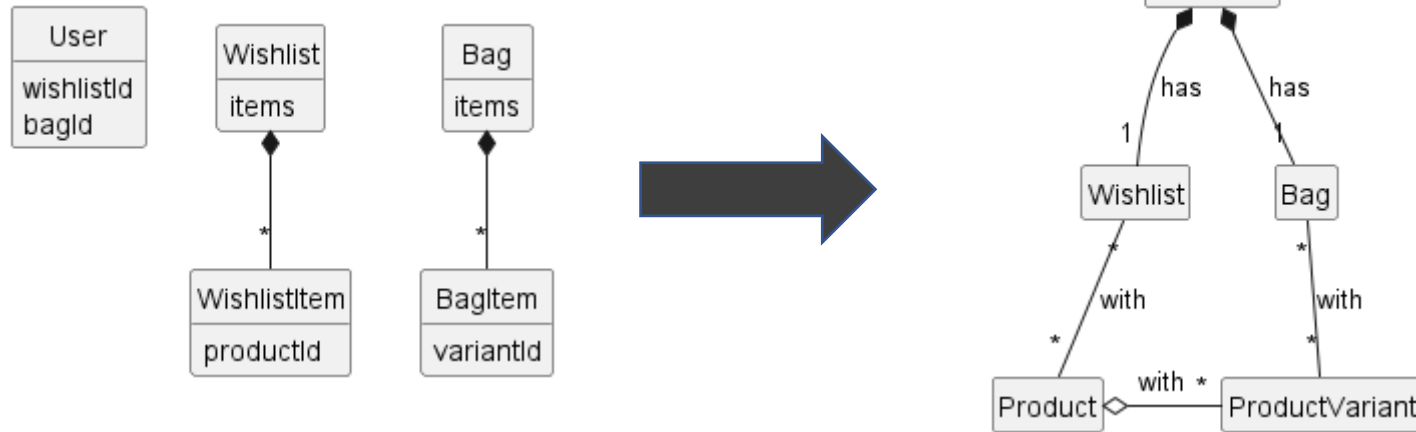
Public APIs enumerate system functionality.

## Evolve technical artifacts into UML-ish business artifacts

Identify entities and relations.

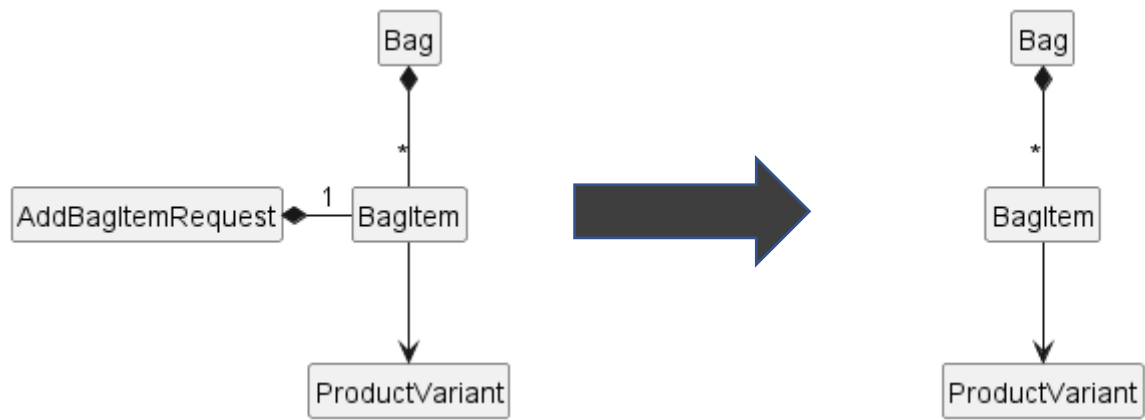
Replace technical identifiers with relationships.

Drop auxiliary entities.



**Guideline: Focus on the end result**

If something evolves during a process,  
focus on the end result.



**Guideline: Replace technical abstractions by their business purpose**

Seek the right level of abstraction.  
Which model is most useful?

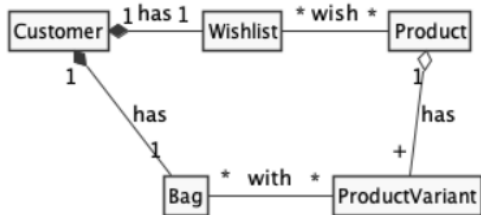


# Bag and Wishlist Domain Model



Created by Joaquim Baptista

Last updated May 02, 2019



## Business definitions

A customer **wish list** remembers products for later consideration by customers.

A customer **bag** collects product variants for check out.

## Business use cases

As a customer,

I want to add a product to my **wish list**  
so that I can remember the product later.

As a customer,

I want to add a product variant to my **bag**  
so that I can purchase the product later.

## Structure of Domain Models at Farfetch

Influenced by information types from Robert Horn:

- UML-ish class diagram (structure).
- Business definitions (concepts).
- Business use cases (procedures),  
with benefits.
- Business process (process)?



## Models have a Social Life

Models typically evolve with discussion, becoming simpler and clearer.

- Work in pairs.
- Always discuss a new model with others.

## DOMAIN MODELS AT FARFETCH

Team of 2-5 for 1000-2000 engineers writers created 28 models between 2018 and 2019.

Takes one day to create a model, more to remove ambiguity.

## Benefits for technical communicators

- Initial conversations with teams gain a clear purpose and objective.
- Writers can argue for the user in business terms, and seek better compromises.
- Tutorials gain business insight and business coverage.
- References gain depth and insight.

# Limitations?

- Models are limited by the APIs being abstracted.  
Hard to maintain a consistent level of abstraction between models.
- SMEs do not engage to maintain models,  
even when they find them useful.
- How to drive consistent conceptual overviews, suitable for partners?

**Describe the context  
to learn the business**

## Systematic top-down approach (2022-)

Business use cases are the best introduction.

Technical experts prefer the UML-ish diagram.

Inspiration comes from an architectural practice:

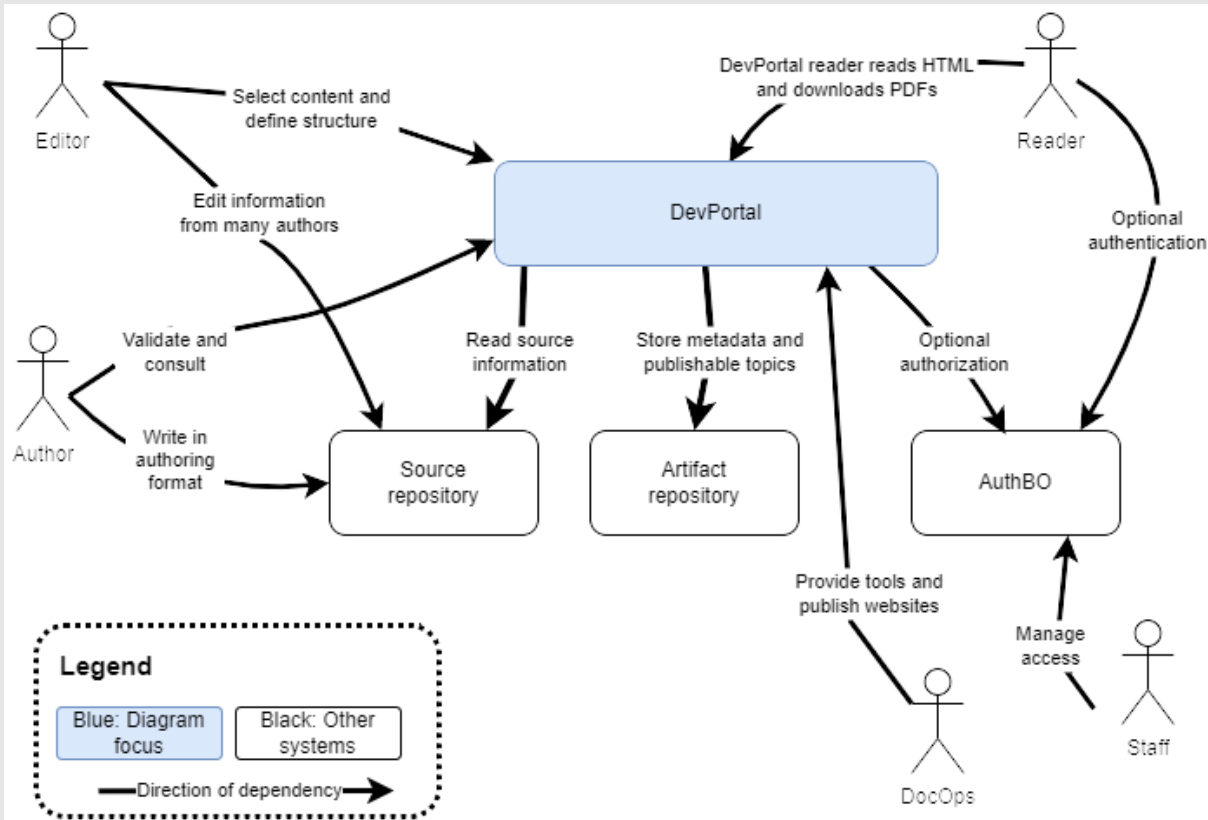
Describes systems based on how the system interacts with its context.

## Contextual diagram

- Treat the system being described as a black box.
- Describe what the system does for people or for other systems.

Complement with text:

- Describe actors.
- Describe actor interactions and their motivations.



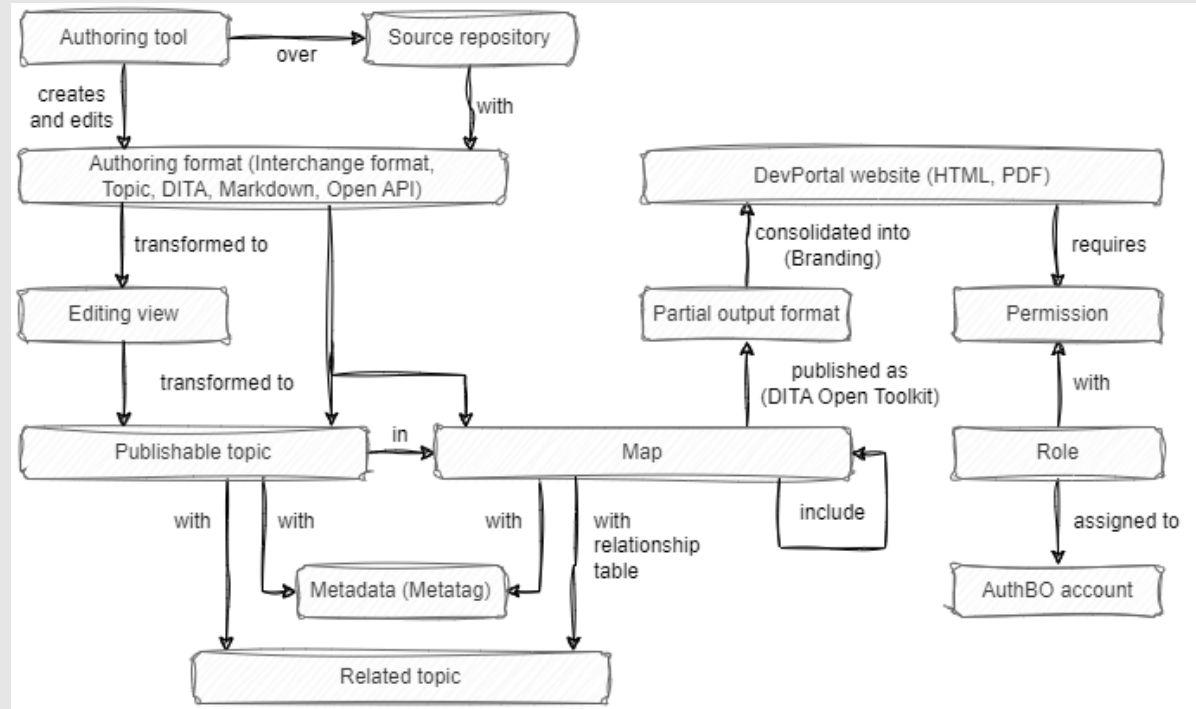
## Business domain model

What do the actors perceive when they interact with the system?

- Identify concepts seen by actors.
- Identify relationships between concepts.

Complement with textual descriptions:

- Describe concepts.
- Describe relationships.

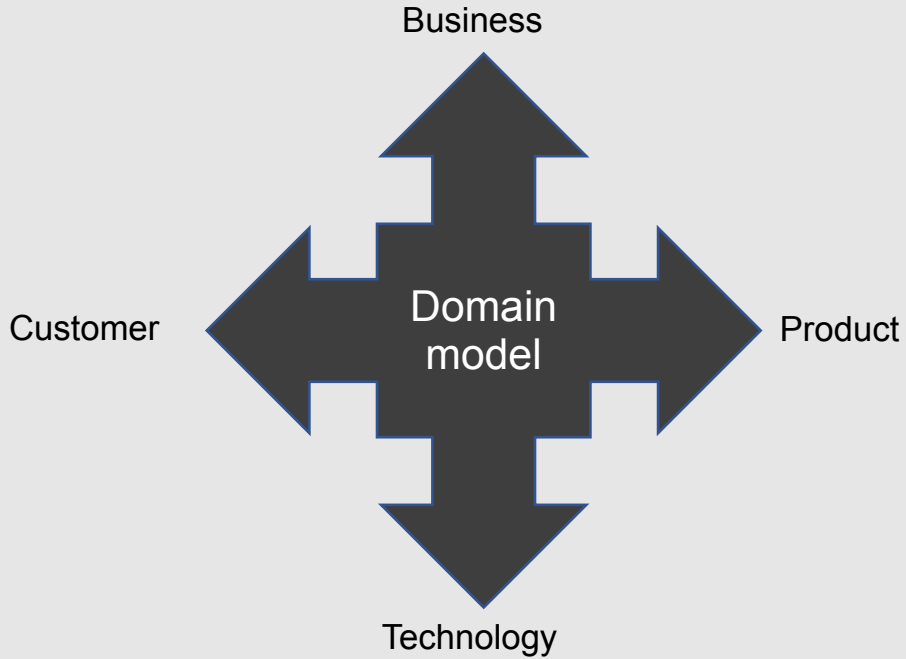


# Expected benefits of contextual approach

- Contextual diagrams are closer to business SMEs.
- Contextual diagrams are consistent if the actors are consistent.
- Business domain models are naturally limited by what actors perceive, driving a consistent level of abstraction.



**Should technical  
communicators use  
domain models?**



What are domain models again?

Description of the product  
and business transactions  
in business terms,  
at an interesting level of detail.

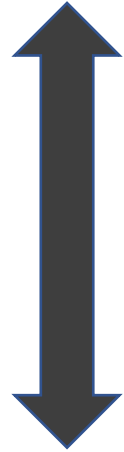
Business impact  
of technology.



Technical impact  
of business  
decisions.

Technical writers are uniquely equipped to  
drive the creation of domain models.

Needs of  
customers.



Customer value  
of product  
features.

## WHAT'S IN IT FOR TECHNICAL COMMUNICATION?

Learn why the product exists.

- Gain leverage to argue with stakeholders.
- Better products, easier to explain.
- Better overviews.

Learn why people use the product,  
and their mental models.

- Better overviews.
- Better tutorials.
- Better insight in docs.

