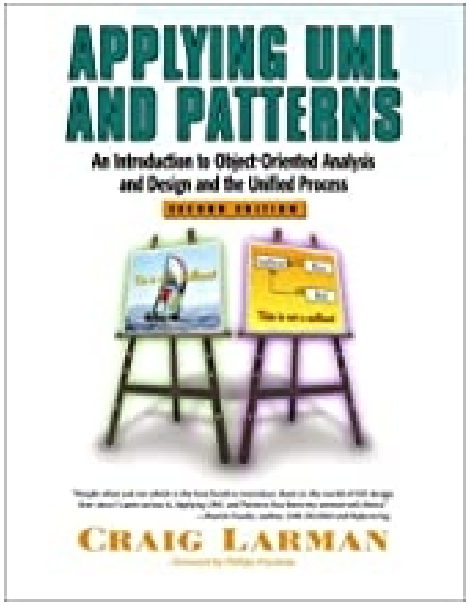
Domain Models

From Larman, Chapter 9

This is one of the most important chapters in the book.

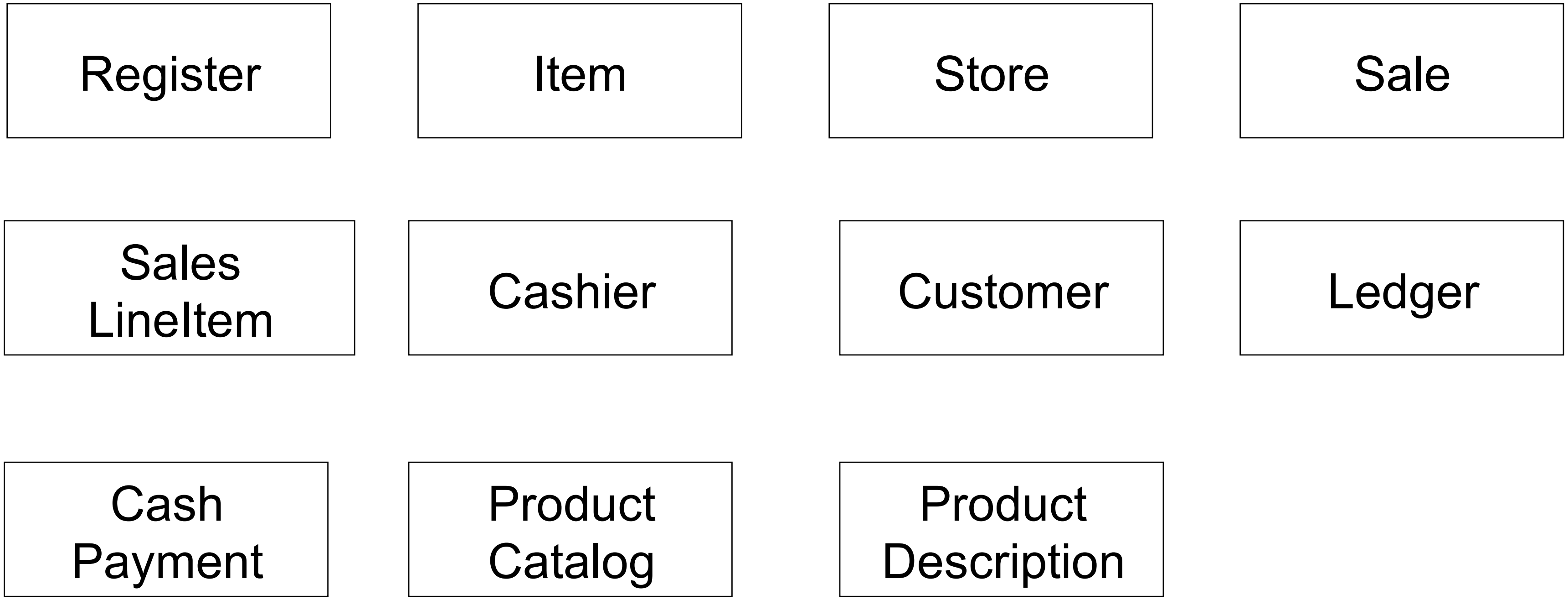

APPLYING UML AND PATTERNS
An Introduction to Object-Oriented Analysis and Design and the Unified Process
SECOND EDITION
CRAIG LARMAN

What is a Domain Model?

- Model of the problem domain, showing concepts, important attributes, and relationships.

- Not a software model.

Concepts in "Make a Sale"
for a Point of Sale (POS) Applicaton



Register

Item

Store

Sale

Sales LineItem

Cashier

Customer

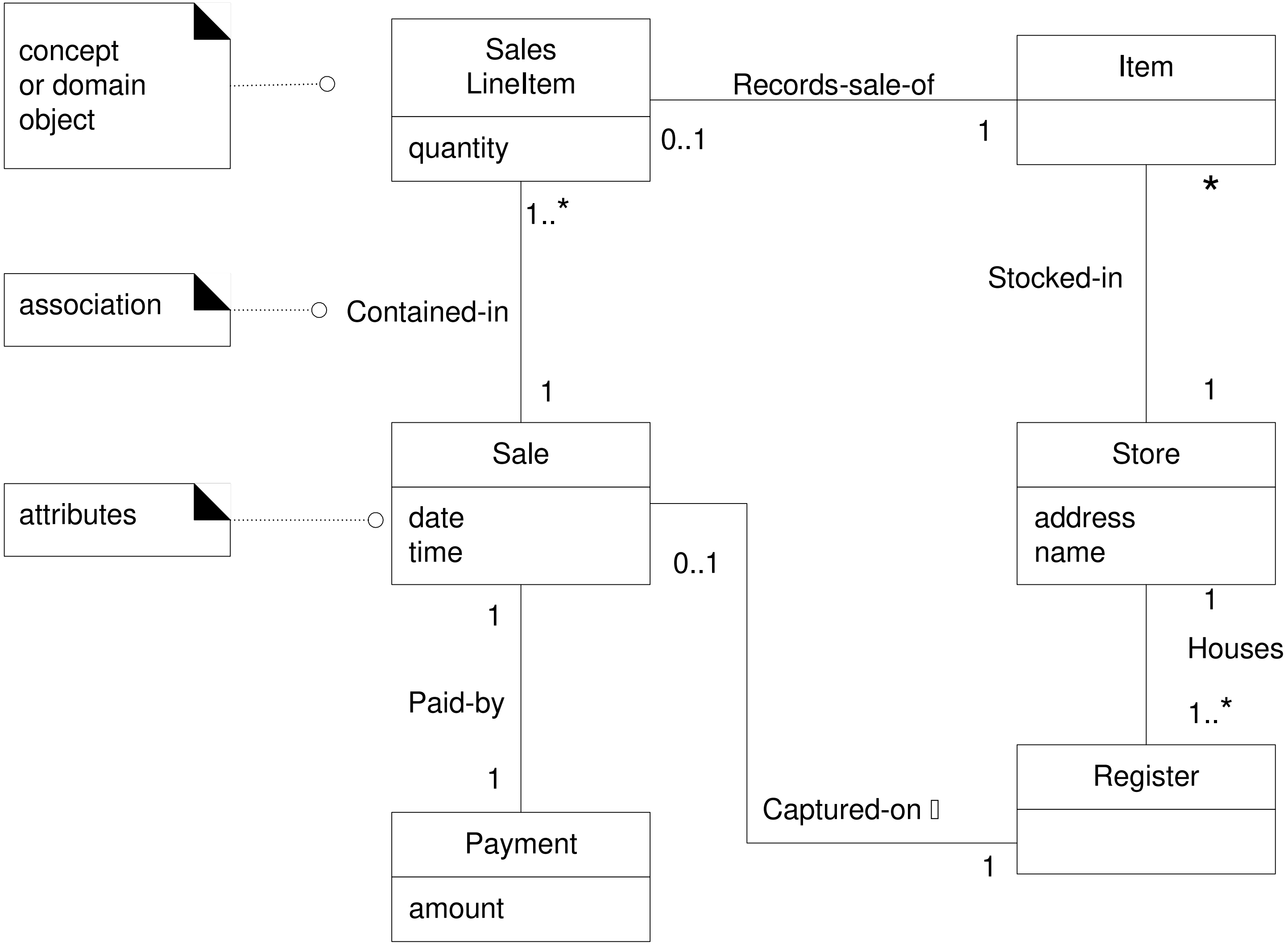
Ledger

Cash Payment

Product Catalog

Product Description

Domain Model for POS based on "Make a Sale"



concept
or domain
object

association

attributes

Sales
LineItem

quantity

Records-sale-of

0..1

1

Item

*

1..*

Contained-in

Stocked-in

1

1

Sale

date
time

0..1

Store

address
name

1

1

Houses

Paid-by

1..*

1

Register

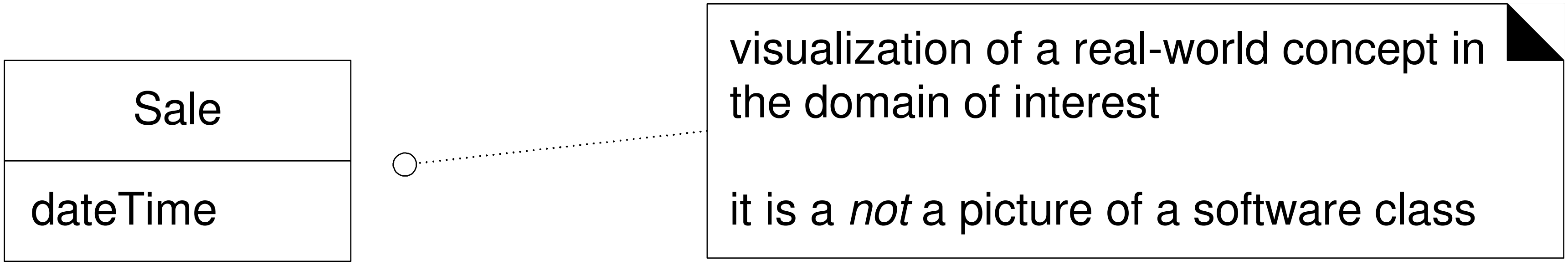
Captured-on

1

Payment

amount

A Domain class



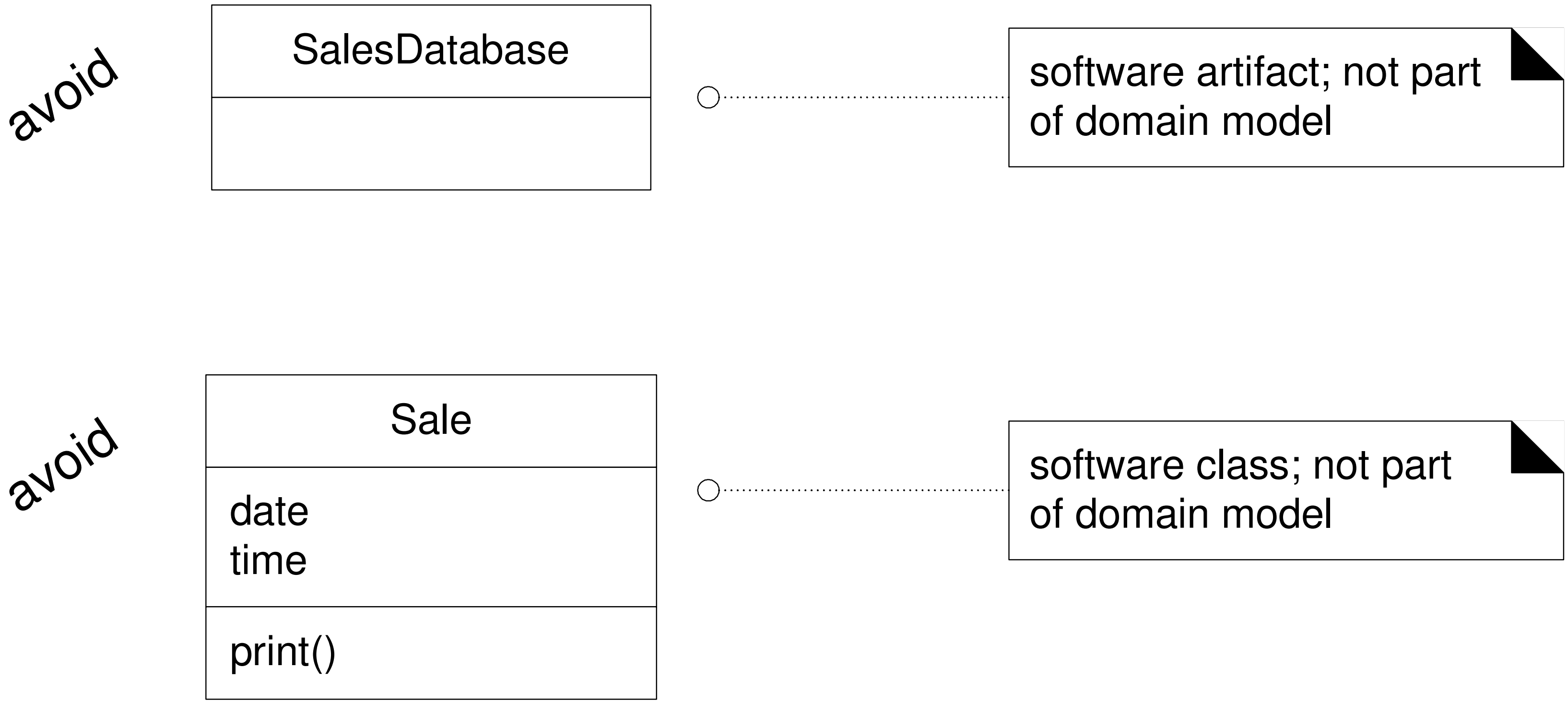
Sale

dateTime

visualization of a real-world concept in the domain of interest

it is a *not* a picture of a software class

Not a Domain level class



avoid

SalesDatabase

software artifact; not part of domain model

avoid

Sale

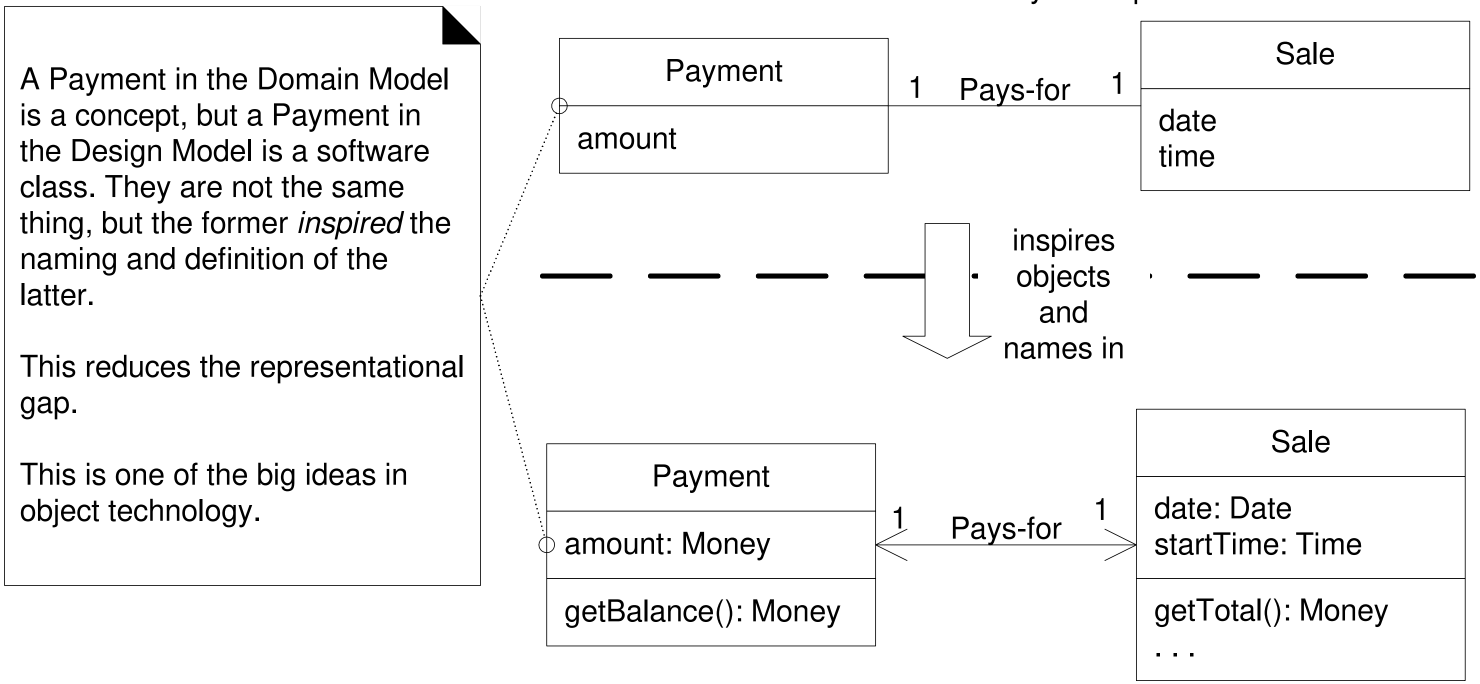
date
time

print()

software class; not part of domain model

Relation between Domain and Design Model

UP Domain Model
Stakeholder's view of the noteworthy concepts in the domain.



A Payment in the Domain Model is a concept, but a Payment in the Design Model is a software class. They are not the same thing, but the former *inspired* the naming and definition of the latter.

This reduces the representational gap.

This is one of the big ideas in object technology.

UP Design Model
The object-oriented developer has taken inspiration from the real world domain in creating software classes.

Therefore, the representational gap between how stakeholders conceive the domain, and its representation in software, has been lowered.

3 Techniques to Discover Domain Classes

1. Look for **noun phrases** in User Stories

2. Use a **category list**

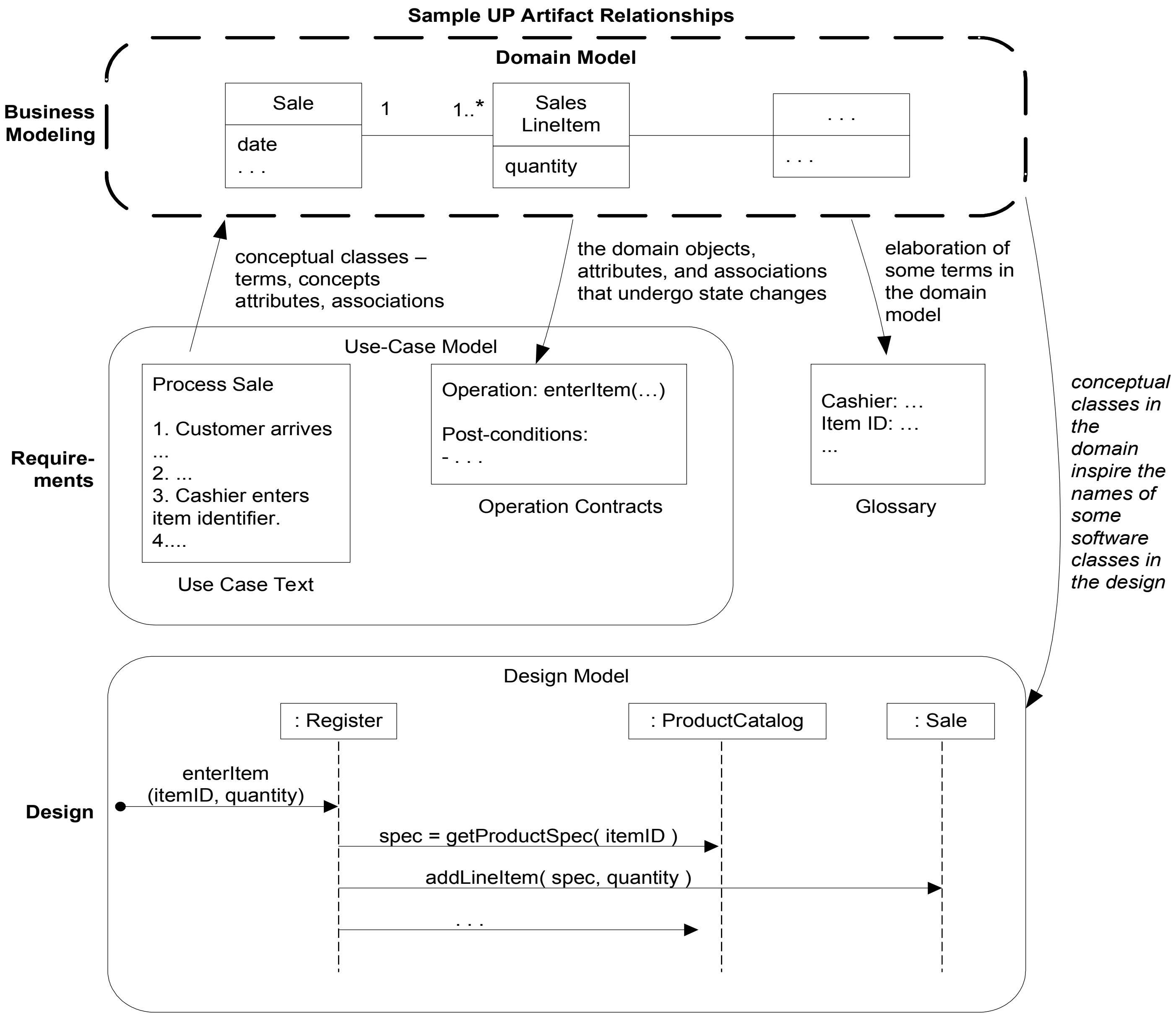
3. Use a similar **existing project**

Conceptual Category List

Table 9.1 on Larman page 140-141.

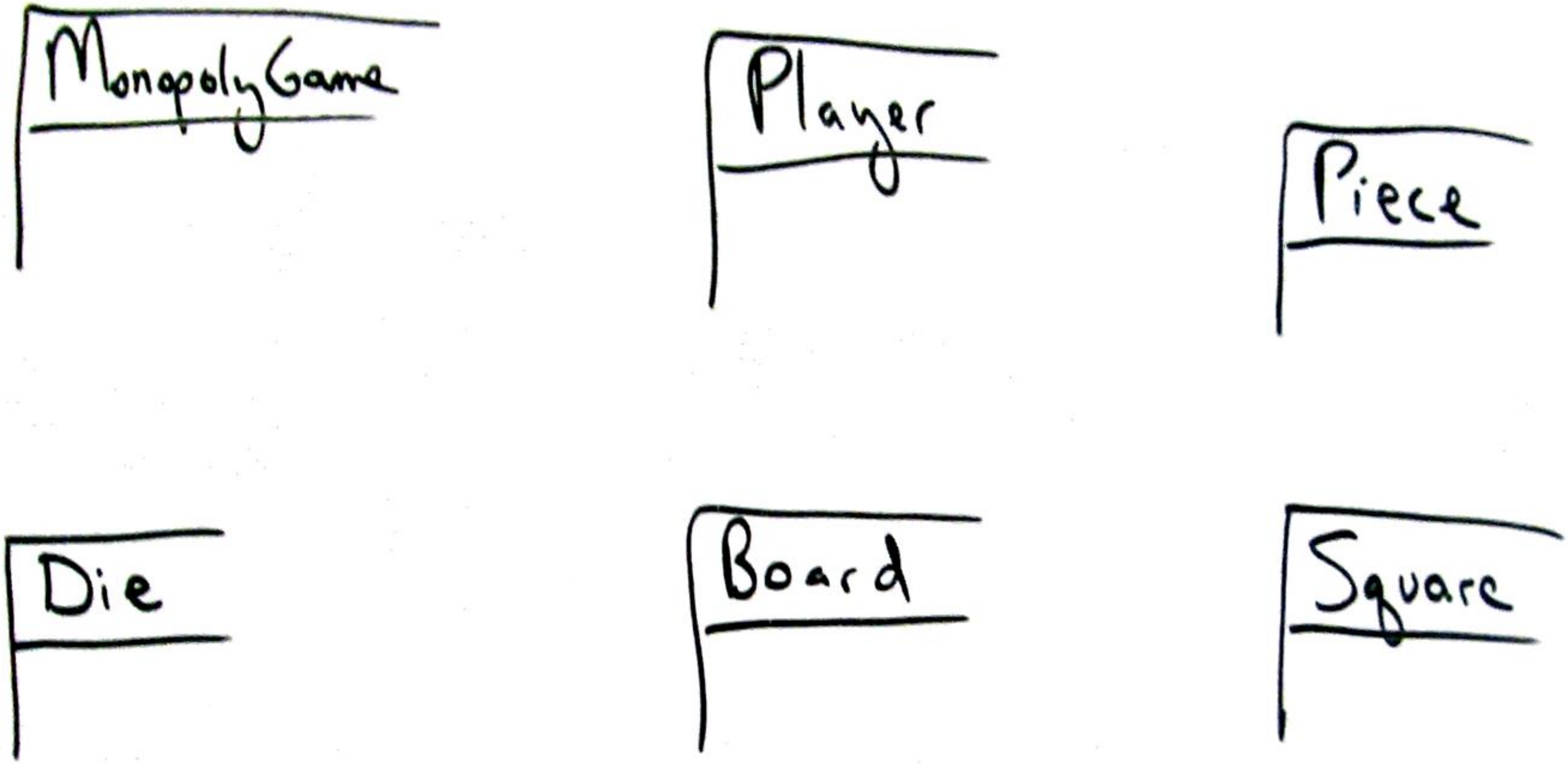
- Business transactions
- Product or service related to transaction
- where is transaction recorded?
- Catalogs
- roles of people related to actors in UC

Design Documents for "Make a Sale"


Sample UP Artifact Relationships
Domain Model
Business Modeling
Sale
date
. . .
1
1..*
Sales LineItem
quantity
. . .
. . .
conceptual classes – terms, concepts attributes, associations
the domain objects, attributes, and associations that undergo state changes
elaboration of some terms in the domain model
conceptual classes in the domain inspire the names of some software classes in the design
Use-Case Model
Requirements
Process Sale
1. Customer arrives
...
2. ...
3. Cashier enters item identifier.
4....
Use Case Text
Operation: enterItem(…)
Post-conditions:
- . . .
Operation Contracts
Cashier: …
Item ID: …
...
Glossary
Design Model
Design
: Register
: ProductCatalog
: Sale
enterItem (itemID, quantity)
spec = getProductSpec(itemID)
addLineItem(spec, quantity)
. . .

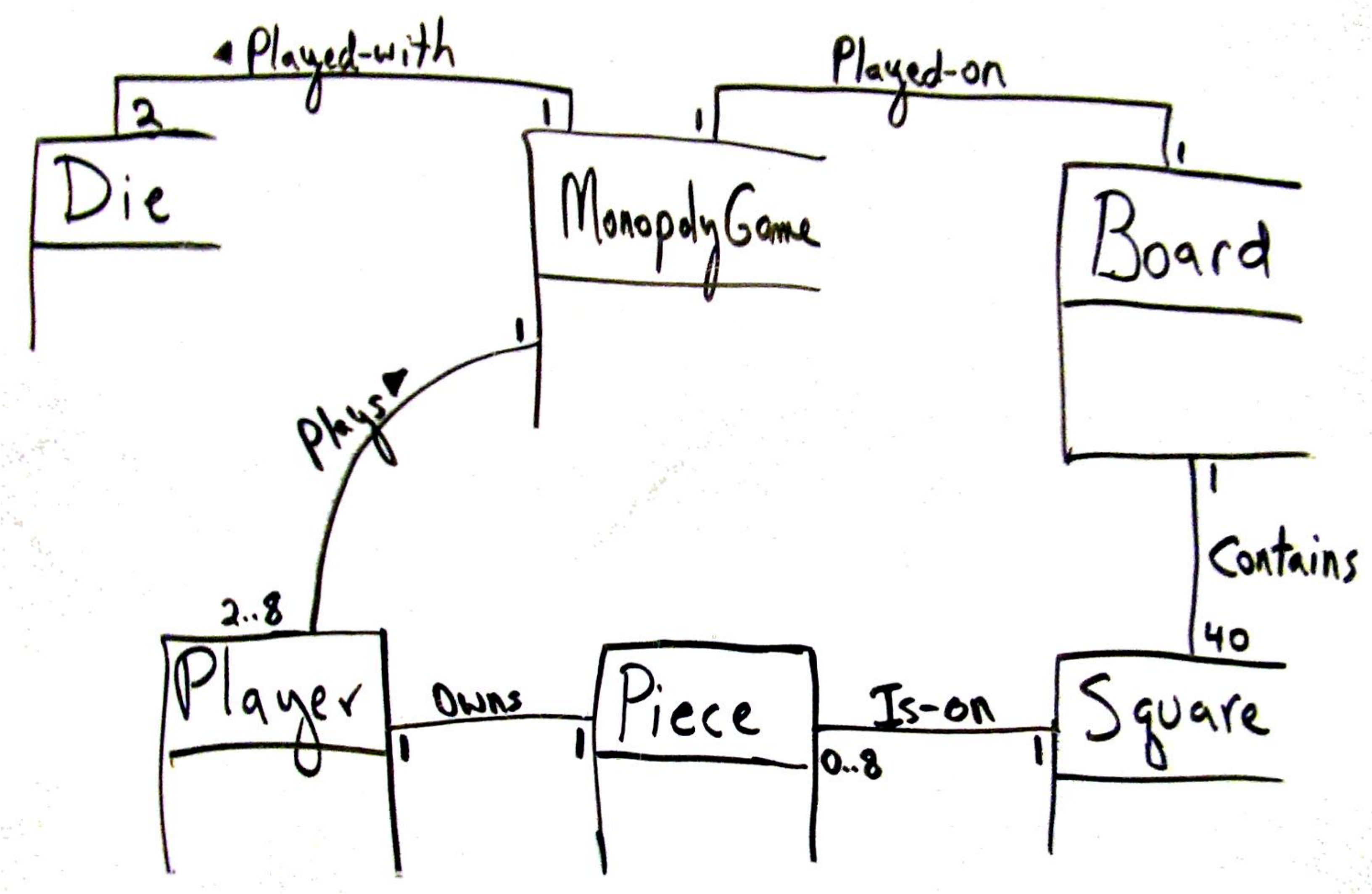
Domain Model for Monopoly Game

Don't try to be beautiful or complete

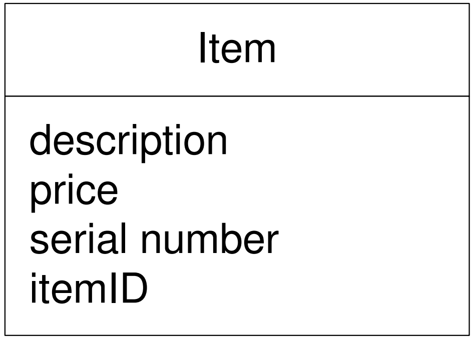


*What is **missing** here?*

Express relationships



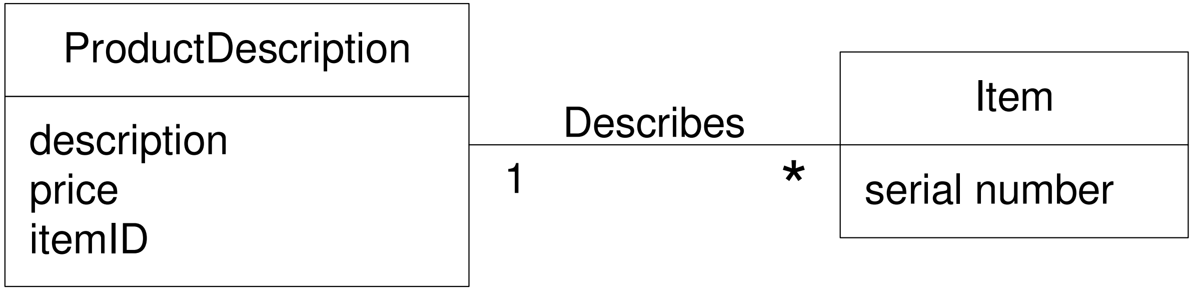
Prefer Associations over attributes



Item

description
price
serial number
itemID

Worse



ProductDescription

description
price
itemID

Describes

1

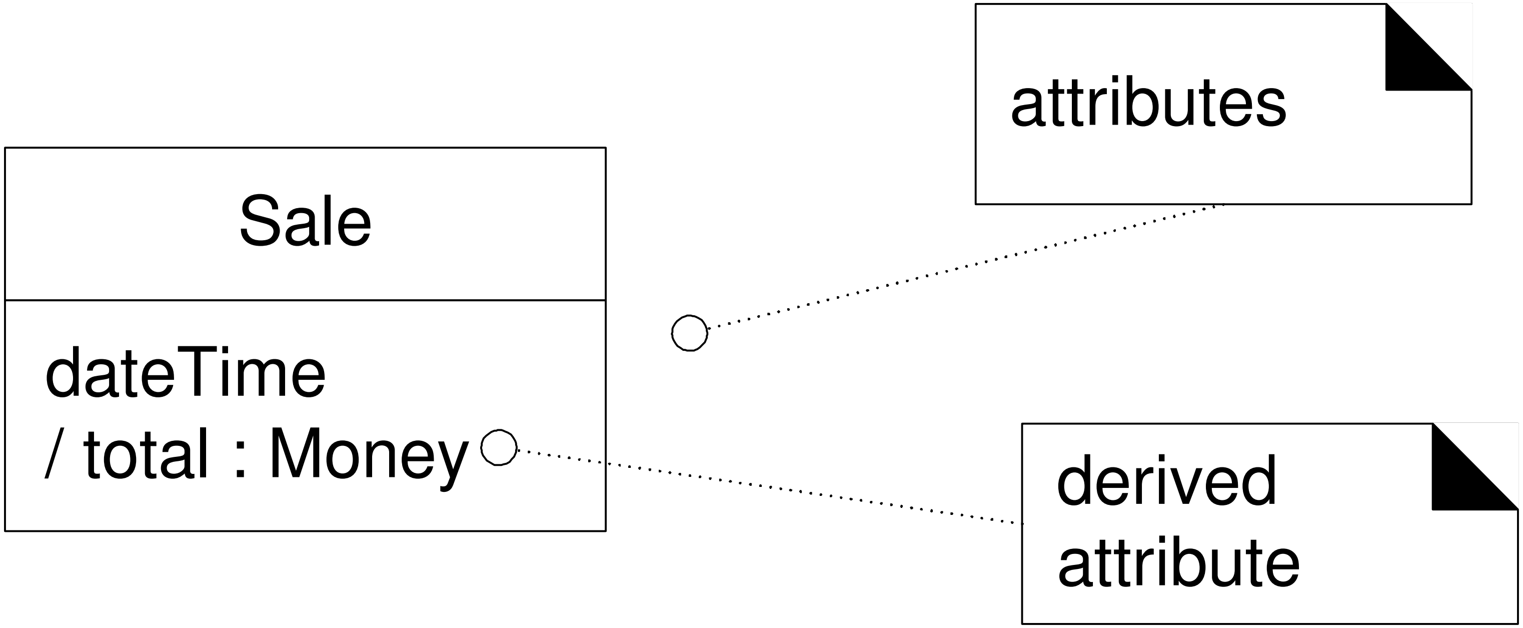
*

Item

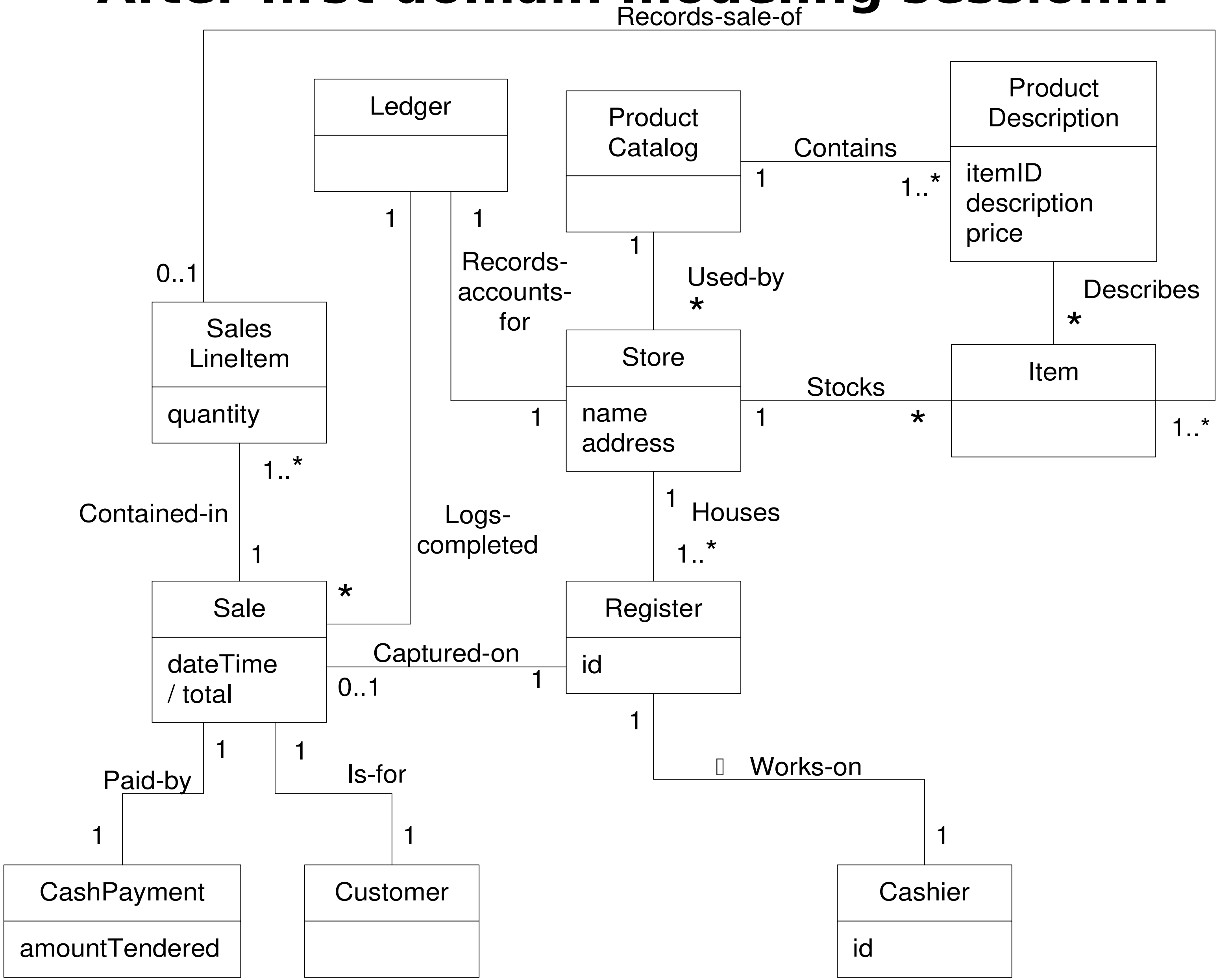
serial number

Better

OK to show "derived" attributes



After first domain modeling session...



Records-sale-of

Ledger

Product Catalog

Product Description

itemID
description
price

Contains

1

1..*

1

1

Records-accounts-for

1

Used-by

*

Describes

*

0..1

Sales LineItem

quantity

Store

name
address

Item

Stocks

1

1

*

1..*

1..*

Contained-in

1

Logs-completed

1

Houses

1..*

*

Sale

dateTime
/ total

Register

id

Captured-on

0..1

1

1

Works-on

1

1

Paid-by

Is-for

1

1

1

CashPayment

amountTendered

Customer

Cashier

id

Your Turn

- Analysis of your Use Cases
- Use a category list
- Don't be influenced by what other teams are doing – design for yourself
- Similar project?