System Analysis and Design

L09. Domain Models

Topics

- Domain Model
- How to Create a Domain Model
- Associations
- Attributes
- Iterative and Evolutionary Domain Modeling

Why Do Domain Modeling?

- A domain model is the most important and classic model in OO *analysis*.
- If you don't understand the domain, you can't program for it effectively
- Domain model lowers the representational gap between mental model and software model

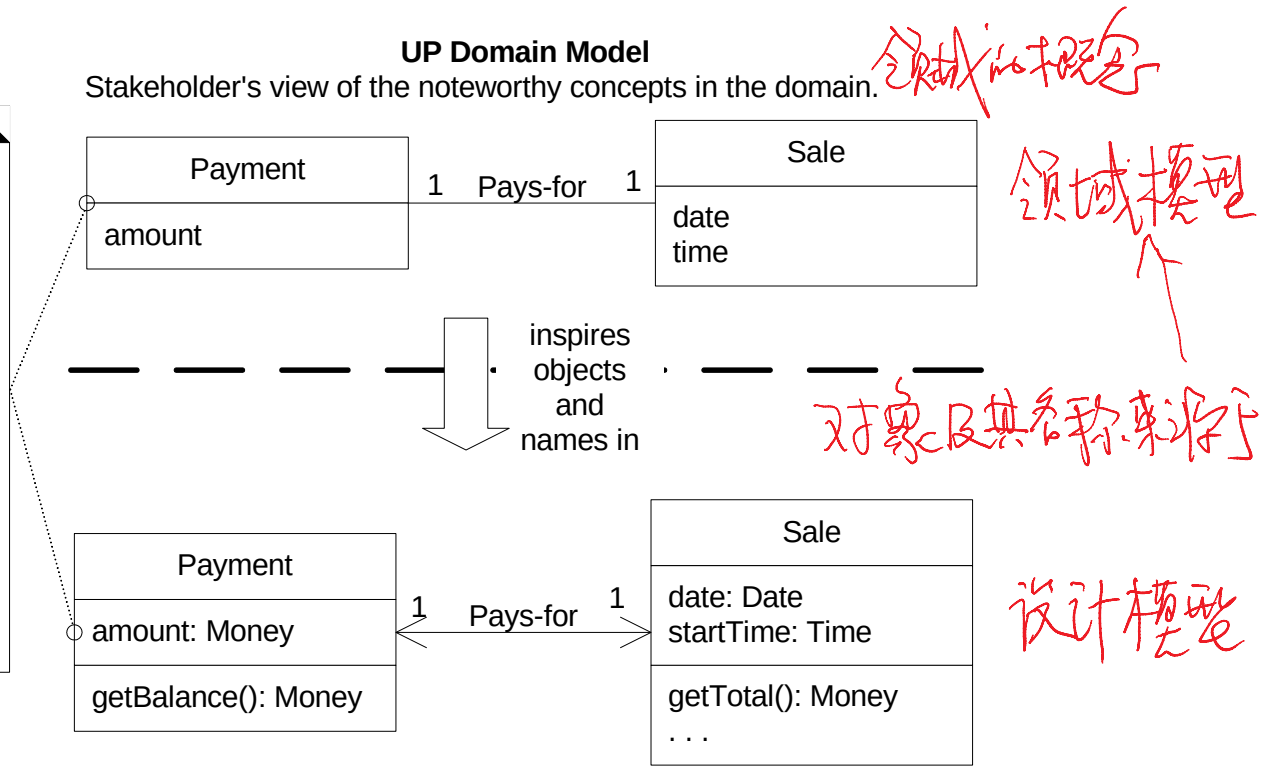
OO idea: 从实际领域的对象到达软件对象.
领域模型使两者的差距变小.

Lower The Representational Gap Between Mental Model and Software Model

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.



Objectives of Domain Modeling

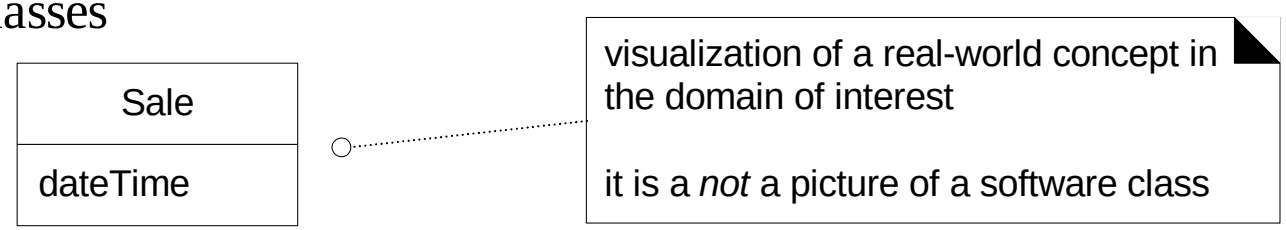
- Identify conceptual classes related to the current iteration
- Create an initial domain model
- Model appropriate attributes and associations in the domain model

What is a Domain Model?

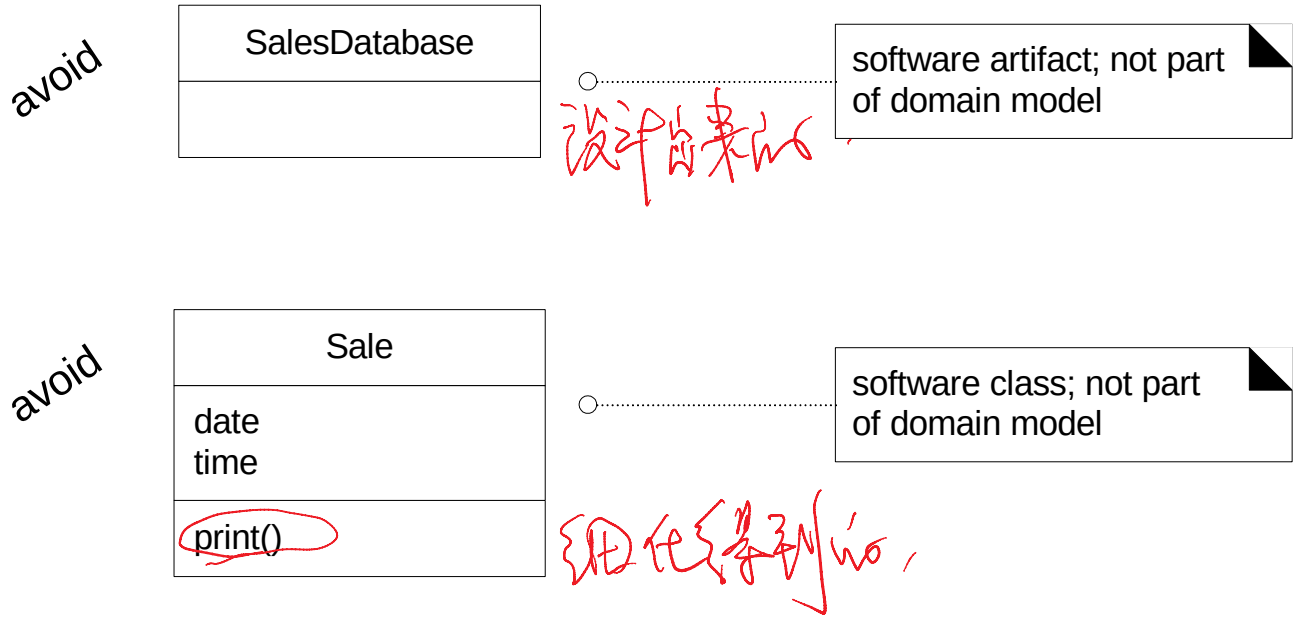
- Illustrates noteworthy concepts in a domain. That is, defines what the system is about
- Models the things in your system and the way they relate to each other
- A domain model is conceptual, not a software artifact
- Domain model can be viewed as a visual dictionary
- Domain models have also been called **conceptual models**, **domain object models**, and **analysis object models**.

What's the Difference?

- A domain model shows real-situation conceptual classes, not software classes

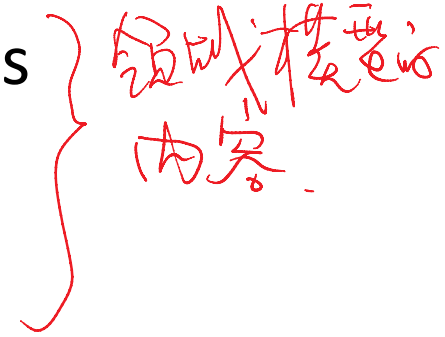


- A domain model does not show software artifacts or classes



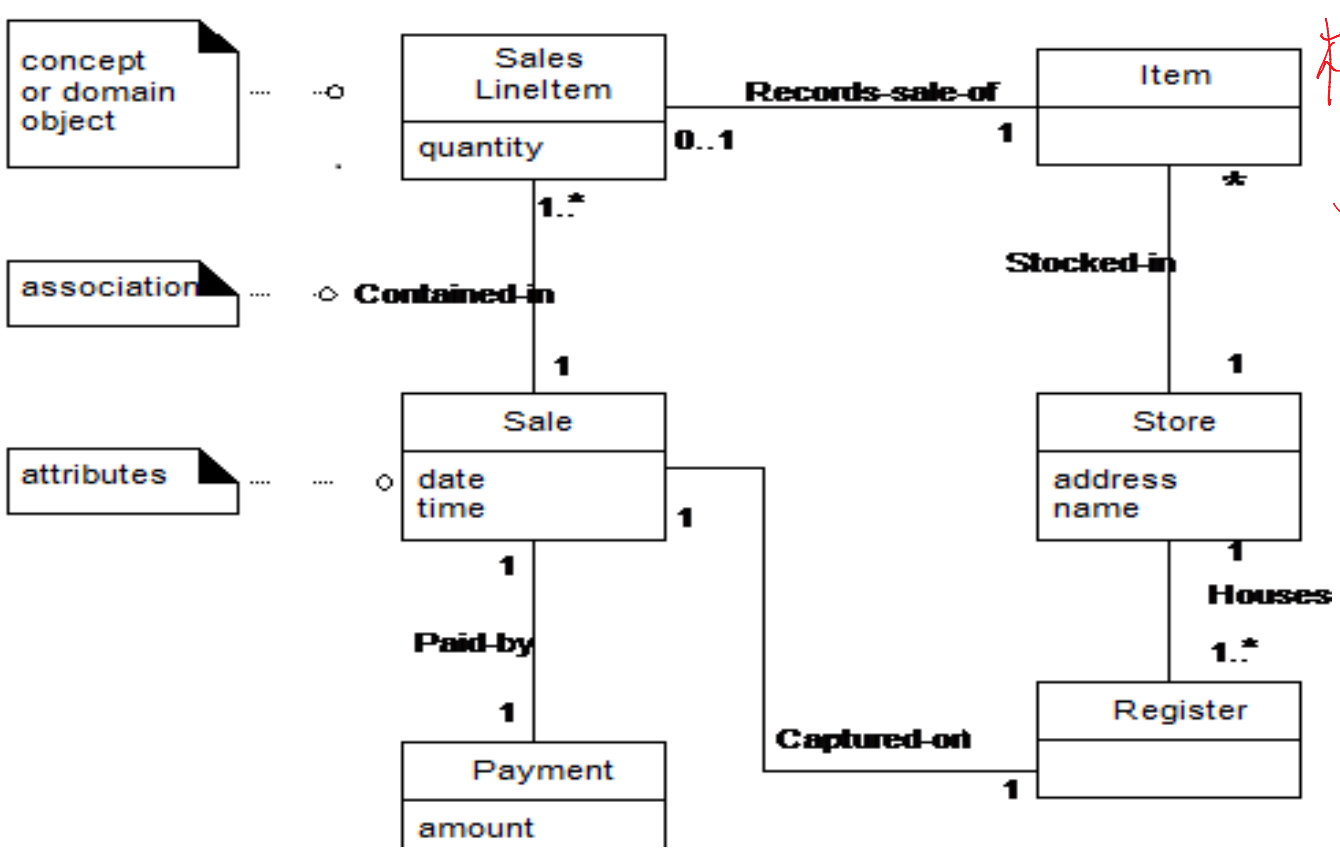
Domain Model and Business Object Model

- The UP Domain Model is a specialization of the UP **Business Object Model** (BOM)
  - BOM focus on explaining 'things' and products important to a business domain
- The UP Domain Model is a set of Class diagrams with no methods, just fields
  - domain objects or conceptual classes
  - Associations between classes
  - Attributes of conceptual classes



Why Call a Domain Model a "Visual Dictionary"?

The information it illustrates could alternatively have been expressed in plain text (in the UP Glossary). But it's easy to understand the **terms** and especially their relationships in a visual language.

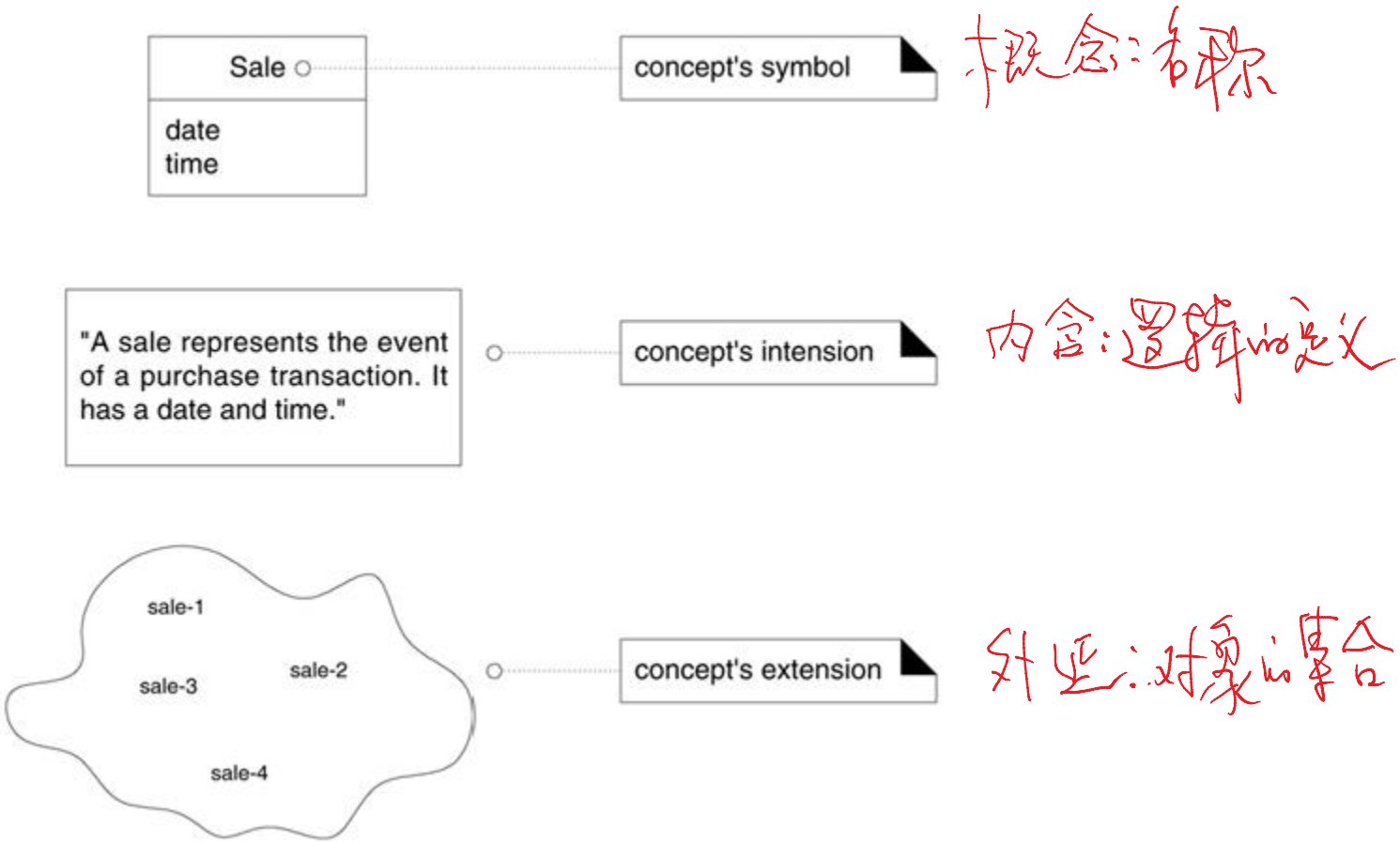


Therefore, the domain model is a ***visual dictionary*** of the noteworthy abstractions, domain vocabulary, and information content of the domain.

What are Conceptual Classes?

- A **conceptual class** is an idea, thing, or object
  - Symbol—Words or images representing a conceptual class
  - Intension—Definition of a conceptual class
  - Extension—The set of examples to which the class applies
- It is not a data model (which by definition shows persistent data to be stored somewhere).
  - It's valid to have attributeless conceptual classes, or conceptual classes that have a purely behavioral role in the domain instead of an information role

A Conceptual Class Has A Symbol, Intension, And Extension



How to Create a Domain Model

- Find the conceptual classes
- Draw them as classes in a UML class diagram
- Add associations necessary to record relationships
- Add attributes necessary for information to be preserved
- Use existing names, the vocabulary of the domain

How do I find Conceptual Classes?

1. Reuse or modify existing models. If there are published models, prior art, or books, use them 重用已有的模型.
2. Use a category list 领域类的分类列表
3. Identify noun phrases (linguistic analysis) 名词分析

Conceptual Class Category List

概念类的分类列表

- We can kick-start the creation of a domain model by making a list of candidate conceptual classes.
- The guidelines in the list also suggest some priorities in the analysis.

Conceptual Class Category 分类	Examples 实例
business transactions Guideline: These are critical (they involve money), so start with transactions.	Sale, Payment Reservation
transaction line items Guideline: Transactions often come with related line items, so consider these next.	SalesLineItem
product or service related to a transaction or transaction line item Guideline: Transactions are for something (a product or service). Consider these next.	Item Flight, Seat, Meal
where is the transaction recorded? Guideline: Important.	Register, Ledger FlightManifest

名词短语识别

Noun Phrase Identification

may come from fully dressed UC Glossary, ...

- Consider the following problem description, analyzed for Subjects, Verbs, Objects:

识别问题描述中的主语、动词、宾语.

The ATM verifies whether the customer's card number and PIN are correct.
S V O O O
If it is, then the customer can check the account balance, deposit cash, and withdraw cash.
S V O V O V O
Checking the balance simply displays the account balance.
S O V O
Depositing asks the customer to enter the amount, then updates the account balance.
S V O V O V O
Withdraw cash asks the customer for the amount to withdraw; if the account has enough cash,
S O V O O V S V O
the account balance is updated. The ATM prints the customer's account balance on a receipt.
O V S V O O

名词短语分析获得：RMAC

Noun Phrases

名词短语

Analyze each **subject** and **object** as follows:

- Does it represent a **person** performing an action? Then it's an actor, '**R**'.
- Is it also a verb (such as 'deposit')? Then it may be a method, '**M**'.
- Is it a simple value, such as 'color' (string) or 'money' (number)?

  Then it is probably an attribute, '**A**'.
- Which NPs are unmarked? Make it '**C**' for class.
- Verbs can also be classes, for example:
  - Deposit is a class if it retains state information

从哪里获得术语？

Where are the Terms?

- Some are in the use case, Glossary,. . .
- Some come from domain experts

- Natural language is imprecise and ambiguous, so you need to use judgment

Find Conceptual Classes: POS

- Main Success Scenario (or Basic Flow):
  - **1.Customer** arrives at a **POS checkout** with **goods** and/or **services** to purchase.
  - **2.Cashier** starts a new **sale**.
  - **3.Cashier** enters **item identifier**.
  - 4.System records **sale line item** and presents **item description**, **price**, and running **total**. Price calculated from a set of price rules.
  - Cashier repeats steps 2-3 until indicates done.
  - 5.System presents total with **taxes** calculated.
  - 6.Cashier tells Customer the total, and asks for **payment**.
  - 7.Customer pays and System handles payment.
  - 8.System logs the completed **sale** and sends sale and payment information to the external **Accounting** (for accounting and **commissions**) and **Inventory** systems (to update inventory).

POS example

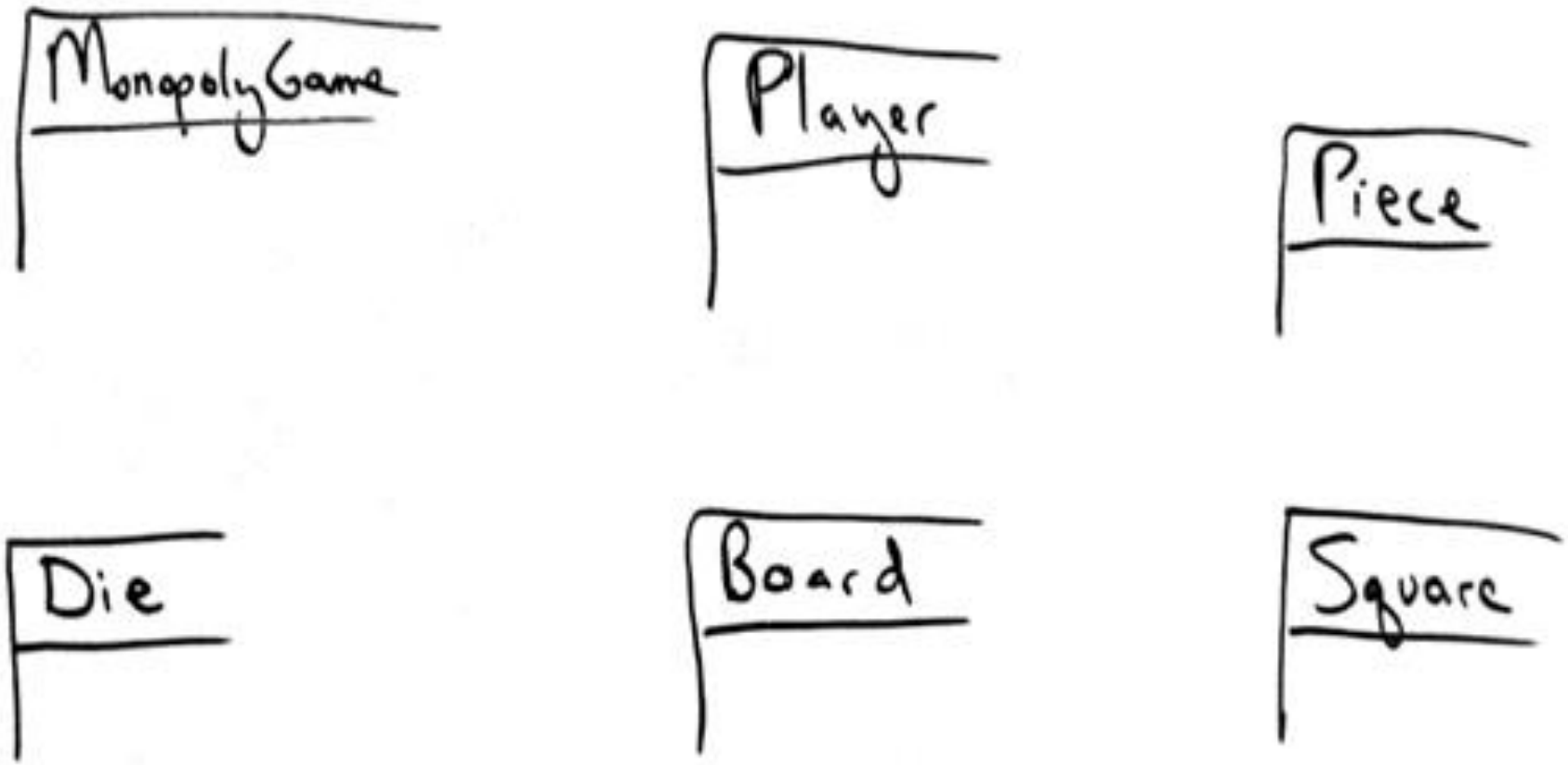
You can create a list, or you can use a set of class diagrams, per the table below

Register	Item	Store	Sale
Sales LineItem	Cashier	Customer	Manager
Payment	Product Catalog	Product Specification	

概念类(领域类)

初始领域模型.

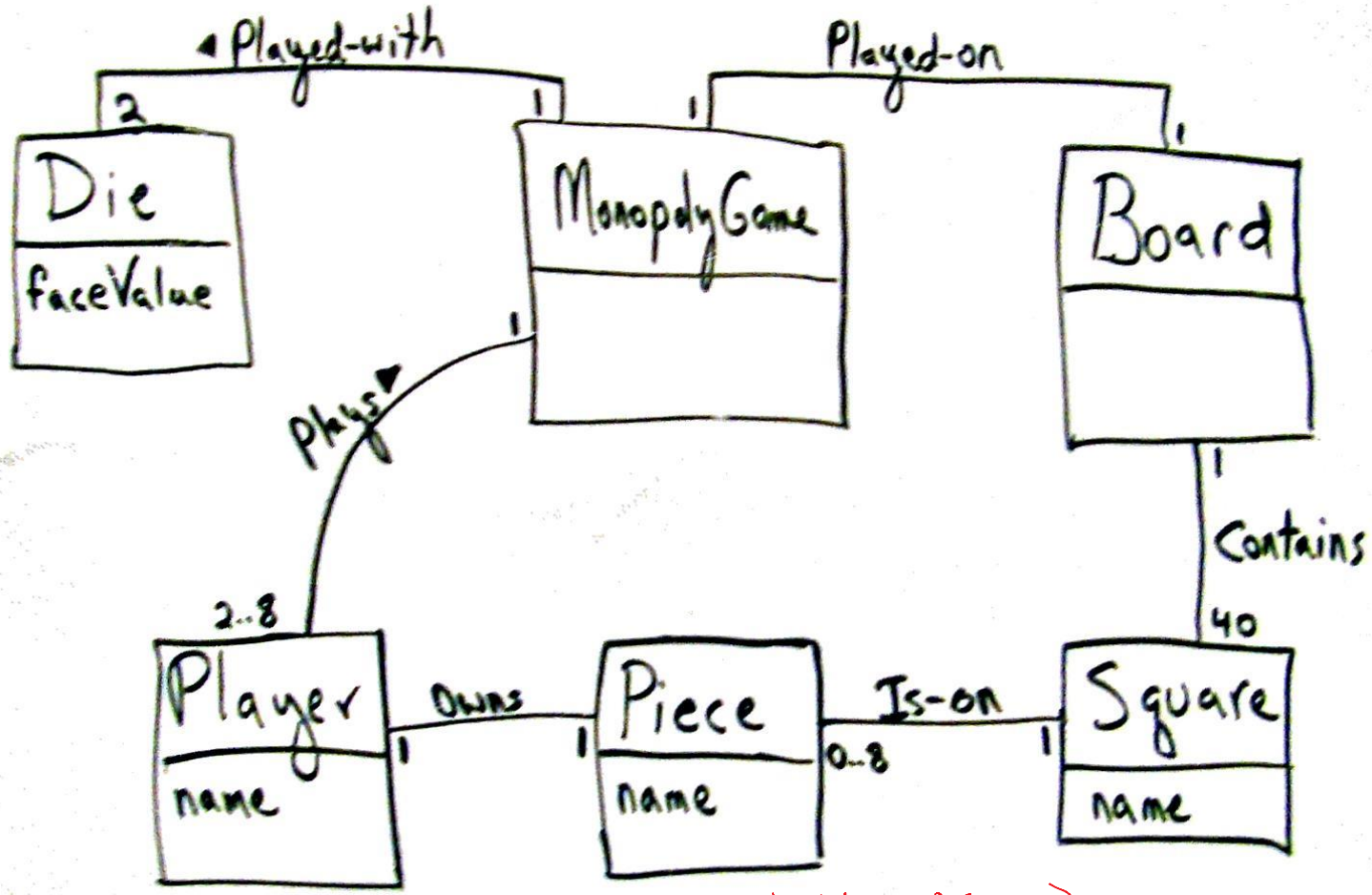
Monopoly Game Domain Model



初始领域模型.

Monopoly Game Domain Model

领域模型例



(已列出了领域类及其关系)

注解.

Maintaining the Model

- Usually the model is a guideline to thinking, not an end in itself. Thus it will change as you learn more
- Who will use the updated model?

如果没有实际的理由，不必维护它的文档。

Report Objects

- Including reports in a domain model usually isn't useful because the information is derived from other objects.
- There are special cases, such as receipts, that should be in the model.

Mapmaker (Domain Terms) Strategy

- Use existing names so far as possible. Learn the terms your users use.
- Exclude out-of-scope features. If a feature is not in the current iteration, don't use it.
- Model the "unreal" world (i. e. Telecom) by listening carefully to the vocabulary of experts. "Port" means two different things in telecom and shipping.

Attributes vs. Classes

- Common mistake: representing something as an attribute when it should have been a class.
- "If we do not think of some conceptual class X **as a number or text** in the real world, it should probably be a class, not an attribute."
- Should "store" be an attribute of *Sale* or should it be a class?

Description Classes

- A description class contains information that describes something else
  - E. g. *Product Description* records the price, picture, text description (and what else?) of an item
- Why Use Them?
  - If we keep all the information in, say, a sales line item, once all of that item are sold, there is no record of what the item was 产品的描述查不到了！
  - How does this relate to database design?

Iterative and Evolutionary Domain Modeling

Discipline	Artifact	Incep.	Elab.	Const.	Trans.
	Iteration	I1	E1..En	C1..Cn	T1..T2
Business Modeling	**Domain Model**		s		
Requirements	Use-Case Model (SSDs)	s	r		
	Vision	s	r		
	Supplementary Specification	s	r		
	Glossary	s	r		
Design	Design Model		s	r	
	SW Architecture Document		s		
	Data Model		s	r	