

Object oriented Analysis &Design

面向对象分析与设计

Lecture_04 Domain Model

Presented by: Xiaohong Chen

2020/9/29

Outline

- Domain models (领域模型)
 - Identify conceptual classes related to the current iteration.
 - Create an initial domain model.
 - Model appropriate attributes and associations.
 - How to get a good quality domain model?

要求掌握的内容

- 如何写好一领域模型?
- 掌握Guidelines
- 掌握两个案例的领域模型

Review: Ch9. Domain Model

- What is a Domain Model?
- A conceptual model, (from Use Case View) NewClass concepts in a problem domain.
- How to visually represent ‘Conceptual class’ in UML?
- In UML it is basically a “class diagram without operations”.
- It may show:
 - Concepts
 - Associations of concepts
 - Attributes of concepts

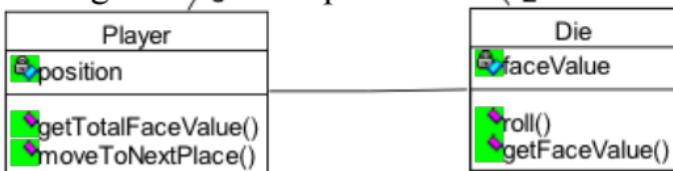
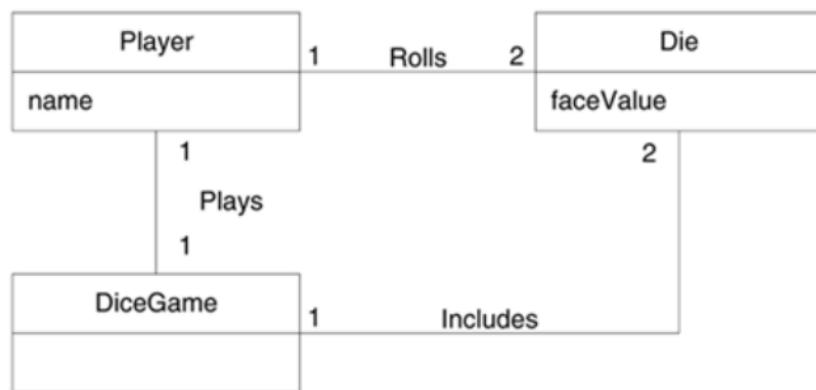


Figure 1.3. Partial domain model of the dice game

A Reference Class Diagram



待思考问题列表

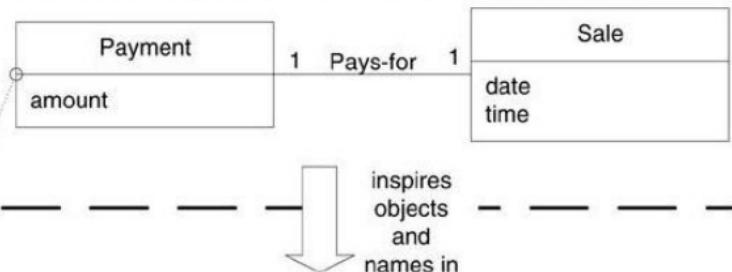
- 领域模型是否等于设计模型？
- 如果不等于，领域模型与设计模型之间的关系？
- 领域模型与数据模型的区别？

Domain Model vs. Design Model

- Same names and notation lower the representation gap

UP Domain Model

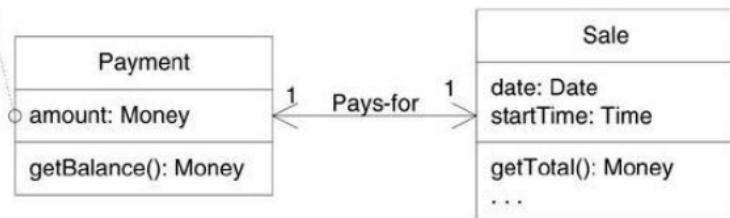
Stakeholder's view of the noteworthy concepts in the domain.



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.

Why domain Model?

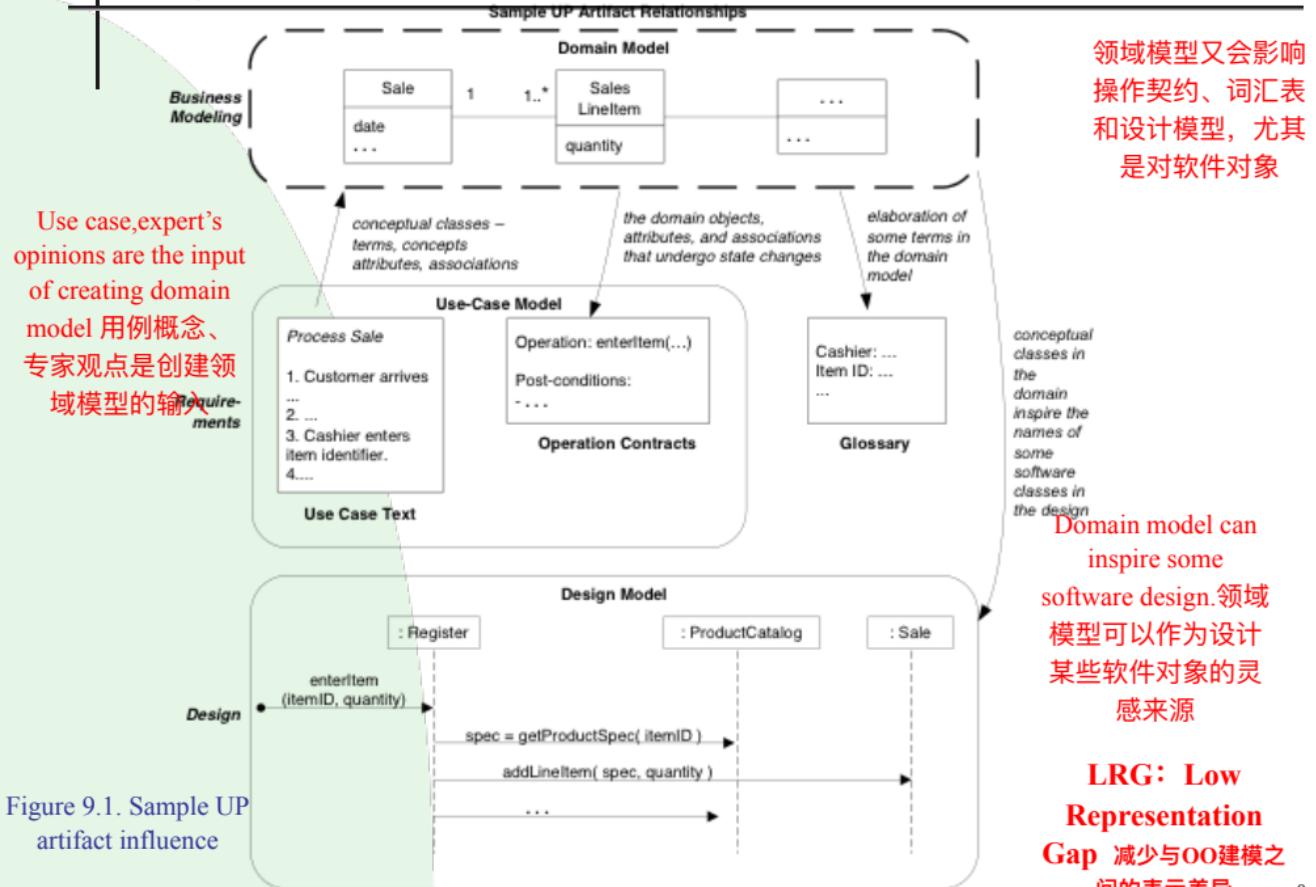


Figure 9.1. Sample UP artifact influence

Review: Domain Model: how to create?

Find conceptual classes

Add associations

Add attributes

What is concept class?

an idea, thing, or object in the problem domain

how to Identify Concepts

Reuse or modify existing models.

This is the first, best, and usually easiest approach, and where I will start if I can.

Finding Concepts with the **Concept Category List**.

Finding Concepts with **Noun Phrase Identification**.

CRC? // stereotype of class [boundary/entity/control]

Domain Model vs. Data Model?

- **Domain Model**

- Concepts, entities, can be abstract

- Relations

- Attributes

- **Data Model**

- How data is stored into persistent storage

- Pure data, in files or databases

Guideline (ch9.7~9.13)

- Agile Modeling -- Sketching a Class Diagram

Monopoly Game

Player

- Agile Modeling-- Maintain the Model in a Tool?
- the purpose of creating a domain model is to quickly understand and communicate of the key concepts.
- Perfection is not the goal
- For tool, may or not

Guideline

- Report Objects / summary object : to be an concept?
 - E.g. Include 'Receipt' in the Model?
 - Factors to consider
 - In general, showing a report of other information in a domain model is not useful since all its information is derived or duplicated from other sources.
 - This is a reason to exclude it.
 - On the other hand, it has a special role in terms of the business rules:
 - It usually confers the right to the bearer of the (paper) receipt to return bought items.
 - This is a reason to show it in the model.

Guideline (ch9.7~9.13)

- Make a domain model in the spirit of how a cartographer or mapmaker works:
 - Use the existing names in the territory
 - For example, if developing a model for a library, name the customer a "Borrower" or "Patron" the terms used by the library staff
 - Exclude irrelevant or out-of-scope features
 - For example, in the Monopoly domain model for iteration-1, cards (such as the "Get out of Jail Free" card) are not used, so don't show a Card in the model this iteration.
 - Do not add things that are not there

Guideline (ch9.7~9.13)

• How to Model the Unreal World?

- Some software systems are for domains that find very little analogy in natural or business domains;
- e.g., software for telecommunications
- Still possible to create a domain model in these domains
- Requires a high degree of abstraction,
 - Stepping back from familiar non-OO designs,
 - Listening carefully to the core vocabulary and concepts that domain experts use.
 - For example, here are candidate conceptual classes related to the domain of a telecommunication switch:
 - Message, Connection, Port, Dialog, Route, Protocol

Guideline

- A Common Mistake with Attributes vs. Classes

Guideline

If we do not think of some conceptual class X as a number or text in the real world, X is probably a conceptual class, not an attribute.(如果我们认为某概念类X不是现实世界中的数字或文本, 那么X可能是概念类而不是属性)

- Use as classes if

Composed of multiple elements (has attributes-states)

There are operations (behavior)

Has units of quantity

- Example,

Store?

consider the domain of airline reservations. Should **destination** be an attribute of Flight, or a separate conceptual class Airport?



Guideline (ch9.7~9.13)

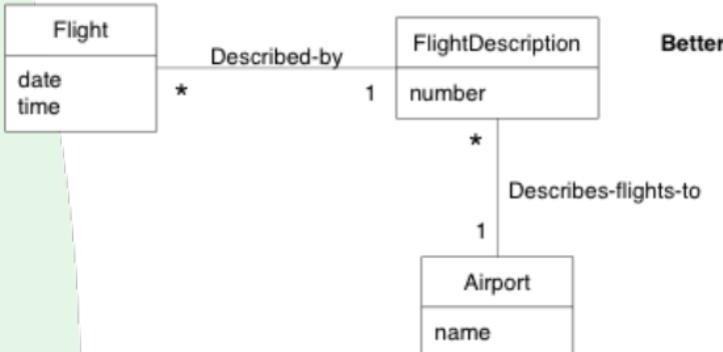
- When to Model with 'Description' Classes?
 - A description class contains information that describes something else.
 - For example, a ProductDescription that records the price, picture, vendor, and text description of an Item
- Why
 - Where to find product information when they were sold out? 商品卖光了之后，到哪里去找商品的信息呢？
 - Each Product of same type has its product description information, is there any problem? 如果每一个同类产品的信息都含有产品描述，有什么问题吗？
 - Repeat, redundant, inconsistency, and a waste of storage space ! 重复、累赘，不一致、浪费存储空间

• For example,



Rules for having Description classes

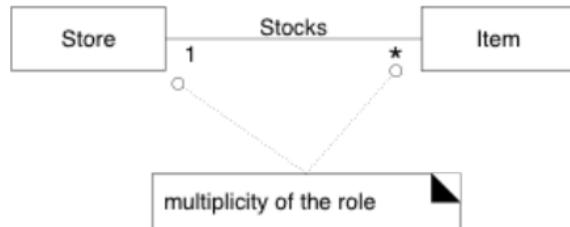
- If descriptions are independent of the existence of items
- If deleting an object deletes information that may need to be maintained
- If it reduces duplications or adds clarity



Associations (review)

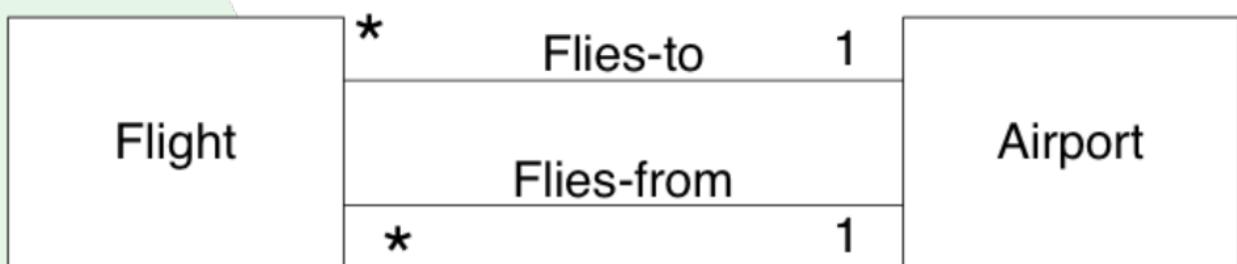
- Relationships between classes
 - Association has a name, (may with direction)
 - Some rules
 - A is contained in or on B: “Board Contains Square.”
 - A owns B: “Players Owns Piece.”
 - A is known in/on B: “Piece Is-on Square.”
 - A is member of B: “Player Member-of (or Plays) MonopolyGame”
 - Multiplicity
 - How many objects participate**
 - Role
 - Multiplicity*
 - Names for the roles of the two classes/objects**
 - Navigability
 - How one object will find the other object to send a message**
 - What is link?

Multiplicity



Multiple Associations

- Two classes can have multiple relations



Guidelines(ch9.14)

Consider including the following associations in a domain model:

- Associations for which knowledge of the relationship needs to be preserved for some duration (“need-to remember” associations).
- Associations derived from the Common Associations List

Why should we avoid adding many associations?

Focus on “need-to-remember” associations).

How to name an association in UML?

Name an association based on a ClassName-VerbPhrase-ClassName format where the verb phrase creates a sequence that is readable and meaningful.

Attributes

An **attribute** is a logical data value of an object

Guideline: When to Show Attributes?

need to remember information

For example,

Sale needs a dateTIme attribute.

Store needs a name and address.

Cashier needs an ID.

The difference of these three attributes?

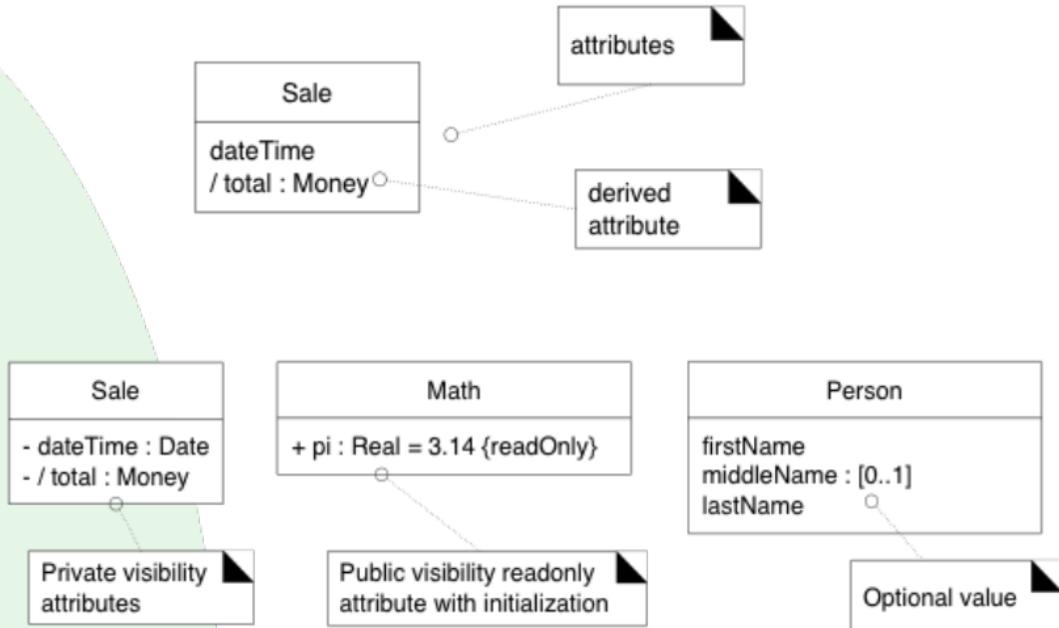
the attribute name with underline means this attribute has class scope 属性名字下面加了下划线，就表示此属性具有类作用域

The attribute name with notation '/' prefix means such attribute is derived attribute 属性名字前面加了'/'，就表示此属性是导出的（Derived Attributes）



DiceGame
numPlayer
/ numDice
name
startGame()
turnControl()
StopGame()

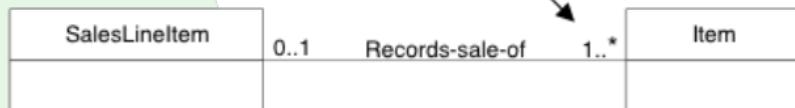
Attributes: notation



Multiplicity vs. derived attribute



Each line item records a separate item sale.
For example, 1 tofu package.



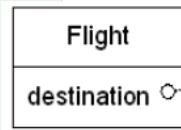
Each line item can record a group of the same kind of items.
For example, 6 tofu packages.



derived attribute from
the multiplicity value

Guidelines

- Two tendencies:



destination is a complex concept

Guideline:

Relate conceptual classes with an association, not with an attribute.



a "simple" attribute, but being used as a foreign key to relate to another object

Guideline:

No attributes representing foreign keys.

Attributes: examples

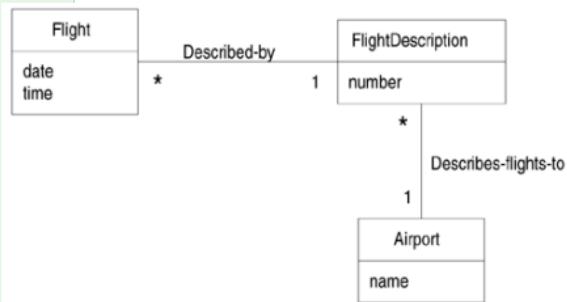
Worse



Better

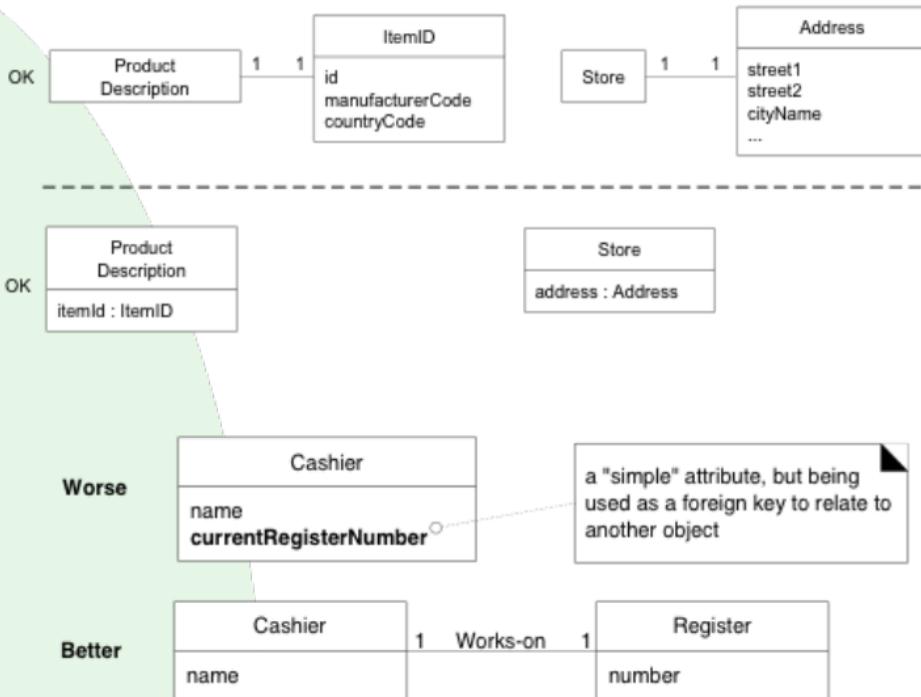


Best



Relate with associations, not attributes.

Attributes: examples

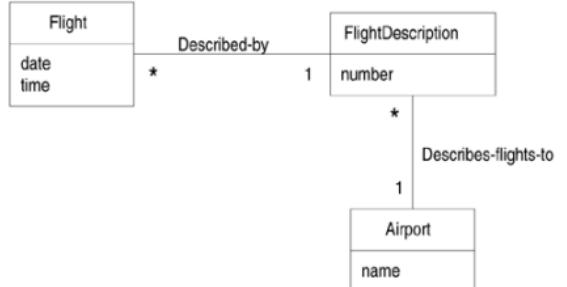


Do not use attributes as foreign keys.

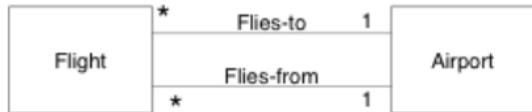
Which one is better?



destination is a complex concept



a "simple" attribute, but being used as a foreign key to relate to another object



作业：

- Construct a conceptual model for the hotel reservation system
- 手绘, 请写出过程, 可以拍照汇总到一个word文档中提交。
- 作业提交日期: 10月13日 大夏学堂



HomeWork

- Read / review text book Ch9 ~14
- Preparation: Review of OOA
- End