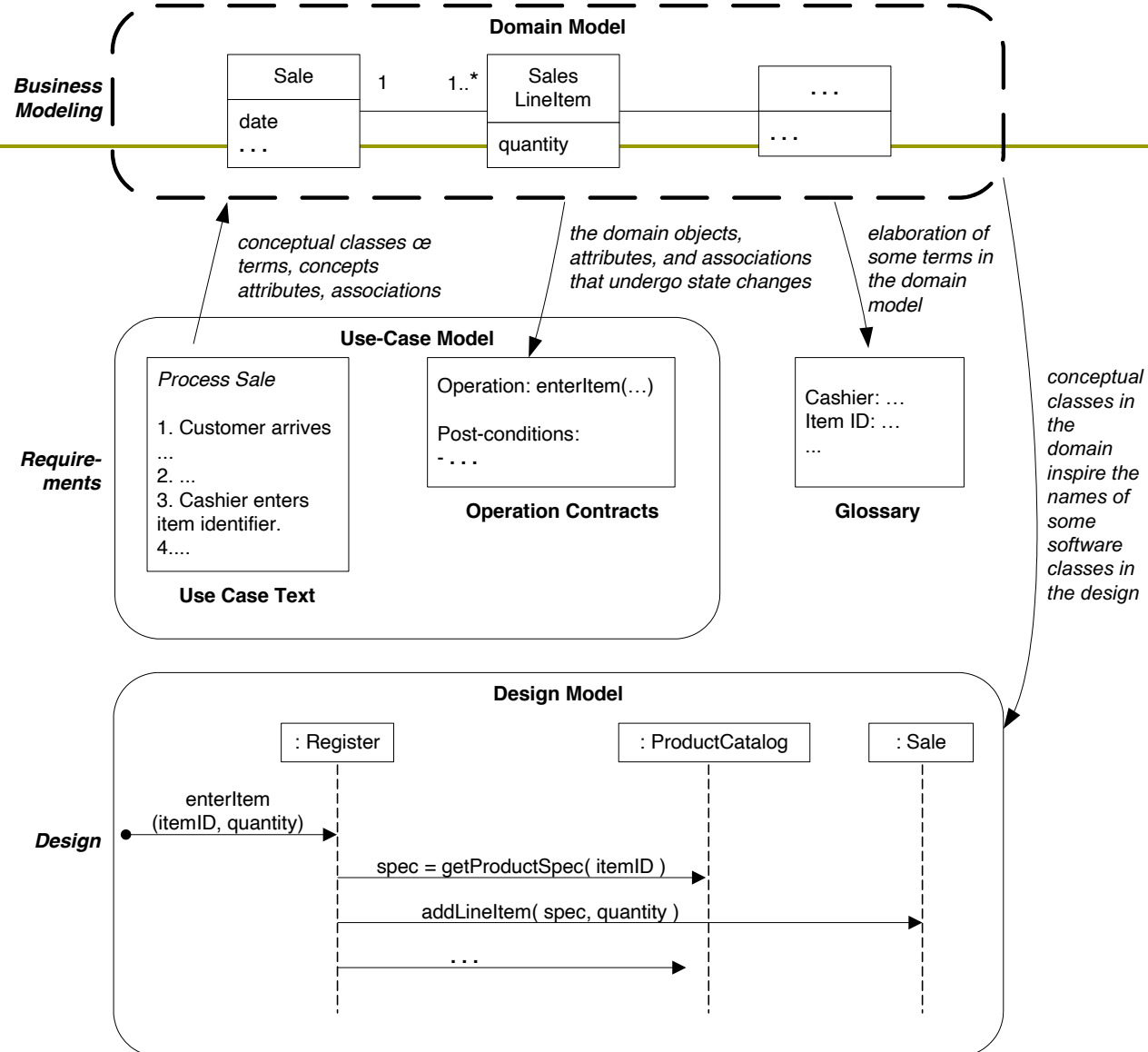


Chapter 9

领域模型

Sample UP Artifact Relationships



目标

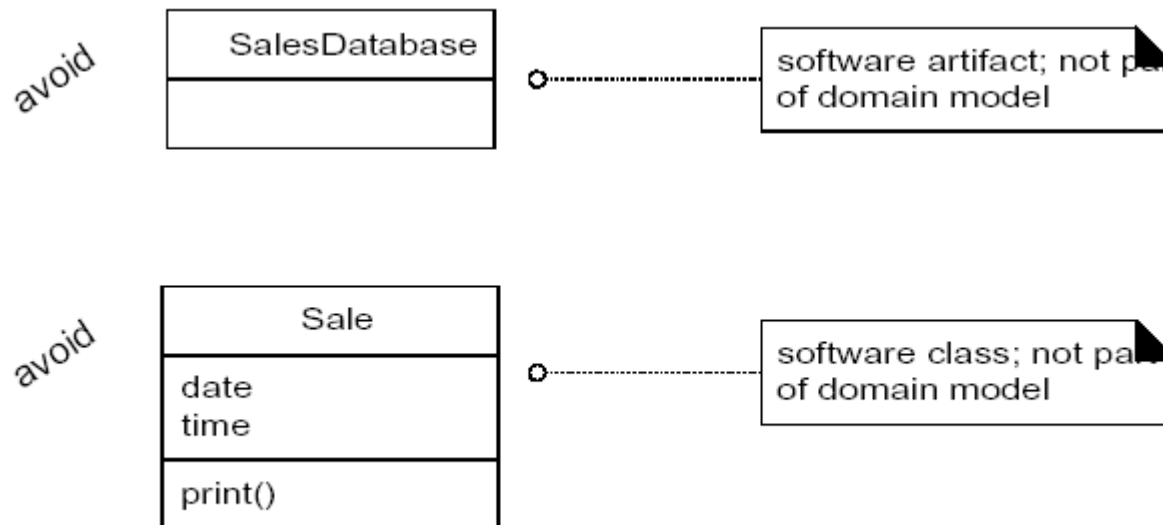
- 处理当前迭代相关的概念类
- 创建初始的领域模型
- 为模型建立适当的属性和关联

定义：什么是领域模型

- 领域模型(Domain Model)是对领域内概念类的可视化表示。
 - 为什么要创建领域模型
提高模型的抽象层次，减少我们的思维和软件模型之间的表示差距。同时促进对领域问题的理解
- 领域模型也称为概念模型、领域对象模型、分析对象模型、可视化字典。
- 通过UML表示法，领域模型被描述为一组没有定义操作的类图，提供了概念透视图。他可以展示：
 - 领域对象或概念类
 - 概念类之间的关联
 - 概念类的属性

领域模型不是软件对象的模型

- 比如，在领域模型中，不会有窗口或数据表之类的对象，也不会有职责或方法



领域模型不是数据模型

- ❑ 数据模型是针对信息系统的模型，用于表示存储于某处的持久性数据。
- ❑ 领域模型考虑的是领域中有哪些概念，数据模型考虑的是这些概念是如何被持久化的。
- ❑ 将领域模型和数据模型混淆是面向对象的开发人员最常见的错误。

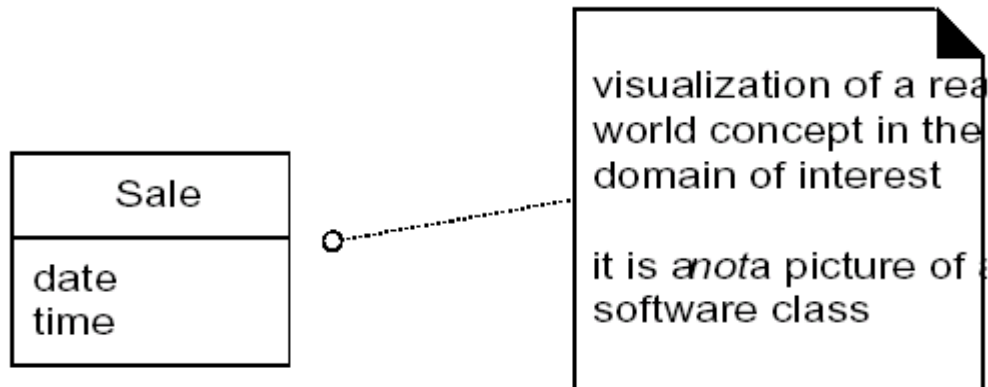
如何创建领域模型

- 寻找概念类
- 将其绘制为UML类图中的类
- 添加关联和属性

概念类

- 领域模型中包含多个概念类
- 概念可以从其符号，内涵和外延来考虑：
 - 符号：表示概念类的词语或图形
 - 内涵：概念类的定义（意义）
 - 外延：概念类的一组示例

概念类的表示法



如何寻找概念类

□ 找到概念类的三条策略

■ 重用和修改现有的模型

□ Martin Fowler的分析模型<analysis patterns>

■ 使用分类列表

■ 确定名词短语

使用分类列表（一）

业务交易	Sale/Payment
交易项目	SalesLineItem
与交易项目相关的产品或服务	Item/Flight/Seat
交易记录的与何处？	Register
与交易相关的人或组织的角色	Cashier/Store
交易/活动地点	Store
重要事件	Sale/Payment
物理对象	Register
事物的描述	ProductDescription
.....

使用分类列表（二）

类别	ProductCatalog
事物的容器	Store/Bin
容器中的事物	Item
其他协作的系统	CreditAuthorizationSystem
金融、工作、合约的记录	Receipt/MaintenanceLog
金融手段	Cash/Check
执行工作所需的进度表、手册、文档	RepairSchedule

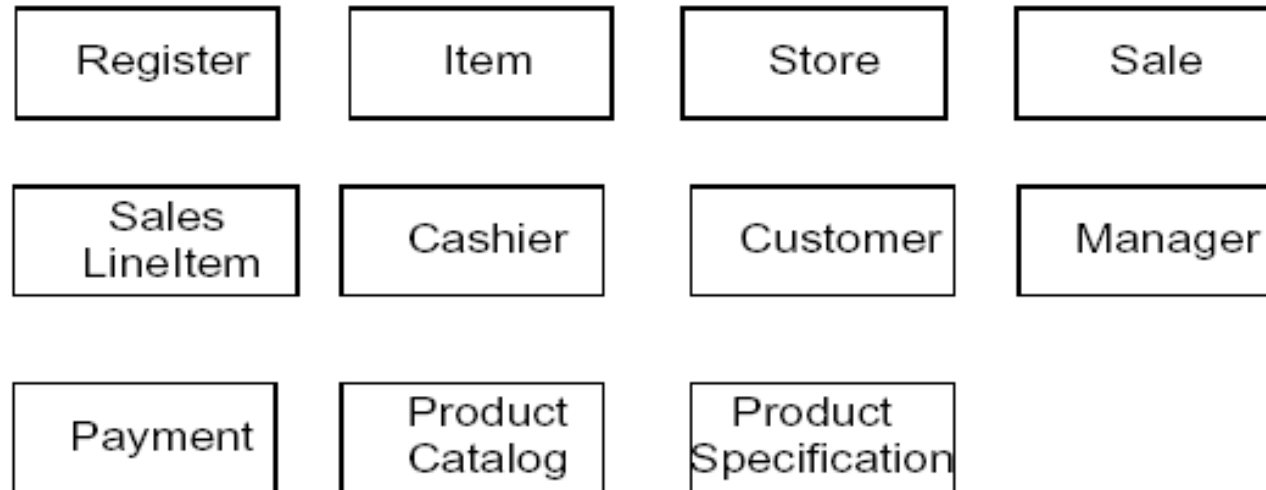
通过识别名词短语寻找概念类

- 通过一种名为语言分析的技术，从对领域的文本性描述中识别名词和名词短语，将其作为候选的概念类或属性。

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).
9. System presents **receipt**.
10. Customer leaves with receipt and goods (if any).

The NextGen POS Domain Model



几个准则

- 敏捷建模：绘制类的草图
- 敏捷建模：是否需要工具维护模型？
- 报表或票据是否是系统领域模型的元素？
- 使用领域术语

描述类

- **Item**的一个实例(instance)代表店铺中一件实际的商品。
- 每一个*Item* 都有 **description**, **price** 和 **Serial number** 的信息, 并且这些信息其他地方没有记录。
- 当一件商品销售出去, 相应的记录会从数据库中删除。

Item
description price serial number itemID

问题

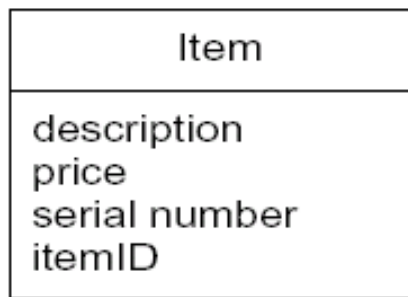
- ❑ 数据重复(description, price, itemID)
- ❑ 如果一件商品销售完毕, 比如ABC Cookie, 谁能够知道这件商品的价格?
- ❑ 如何处理价格的变动?

Item
description price serial number itemID

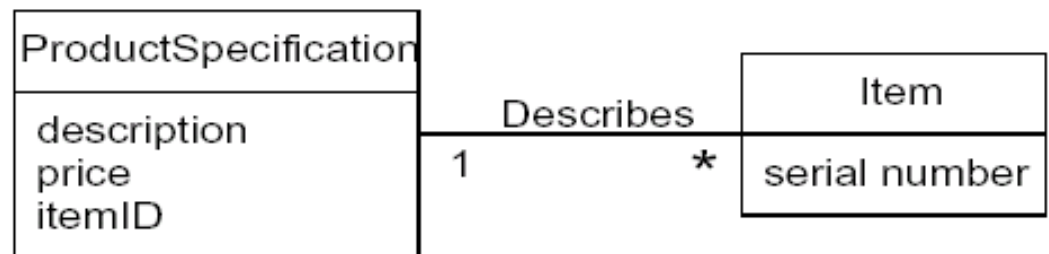
解决方案：使用描述类

- 引入一个新的概念： *ProductSpecification*，代表了一类商品的信息。

Worse



Better

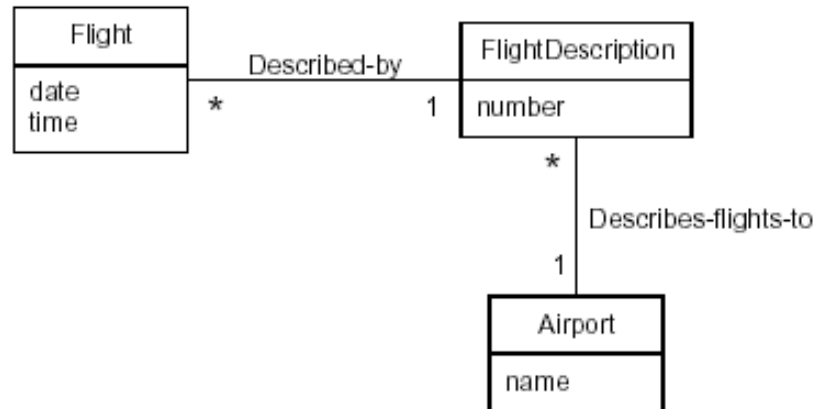


航班

Worse



Better



关联

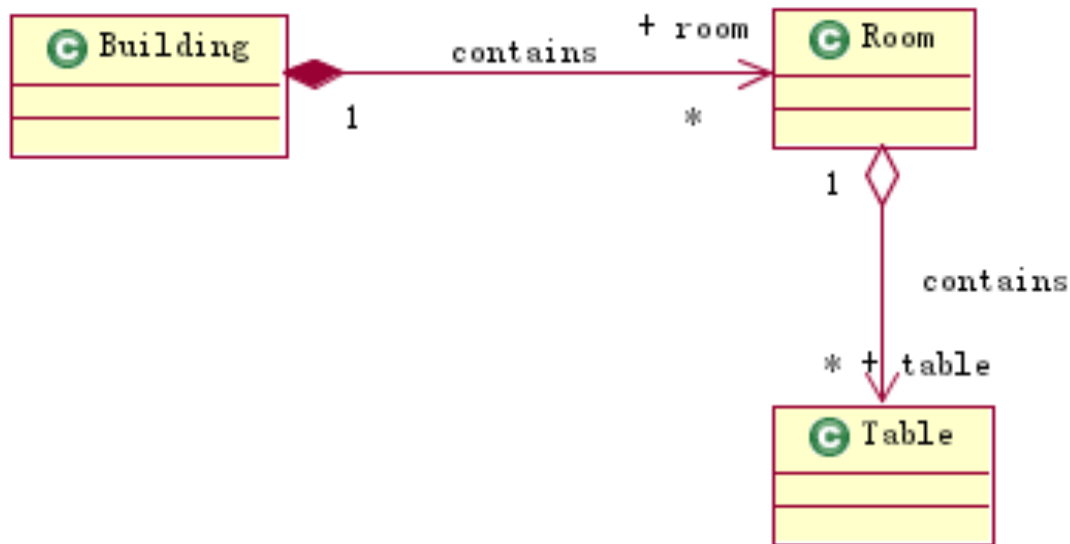
- ❑ 概念不是孤立存在的，他们之间存在各种各样的关系。
- ❑ 关联是类之间的关系，表示有意义的，值得关注的连接。
- ❑ 避免表现出大量的关联，连线太多会产生“视觉干扰”，使图变得混乱。
- ❑ 避免考虑软件的实现，添加关联是为了突出我们对重要关系的大致理解，而非考虑数据结构。
 - 在软件中实现关联是实现领域模型中最复杂的部分之一。

常用的关联列表

A是与交易B相关的交易	<i>CashPayment-Sale</i>
A是交易B中的一个项目	<i>SalesLineItem – Sale</i>
A是B的物理或逻辑组成部分	<i>Drawer-Register</i> <i>Seat-Airplane</i>
A被物理或逻辑地包含在B中	<i>Register-Store</i>
A是B的描述	<i>ItemDescription – Item</i>
.....

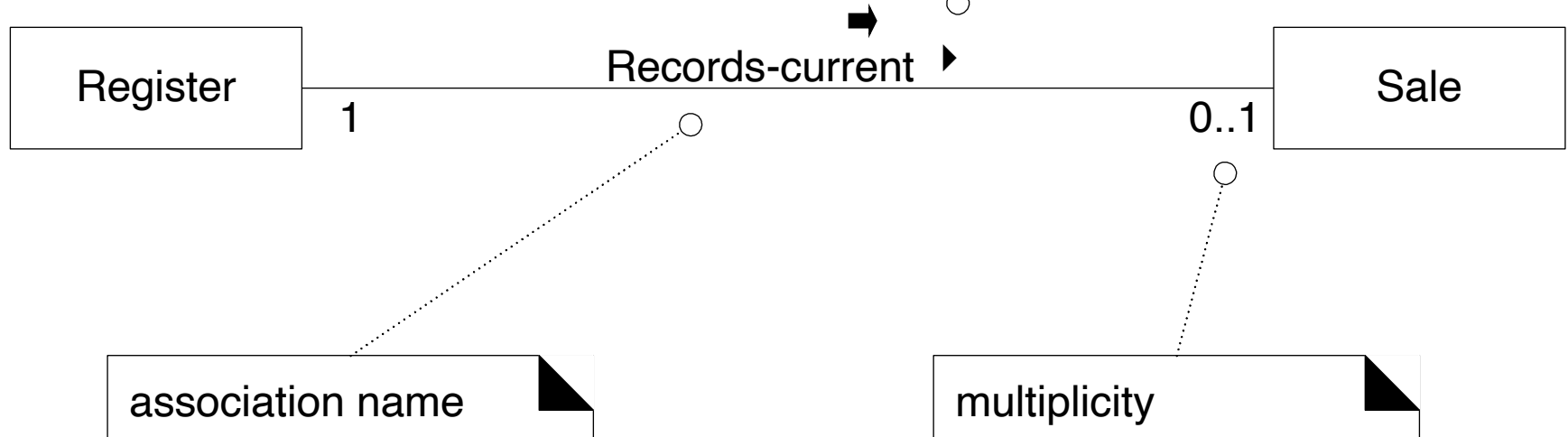
最常见的关联

- A是B的物理或逻辑组成部分
- A被物理或逻辑地包含在B中



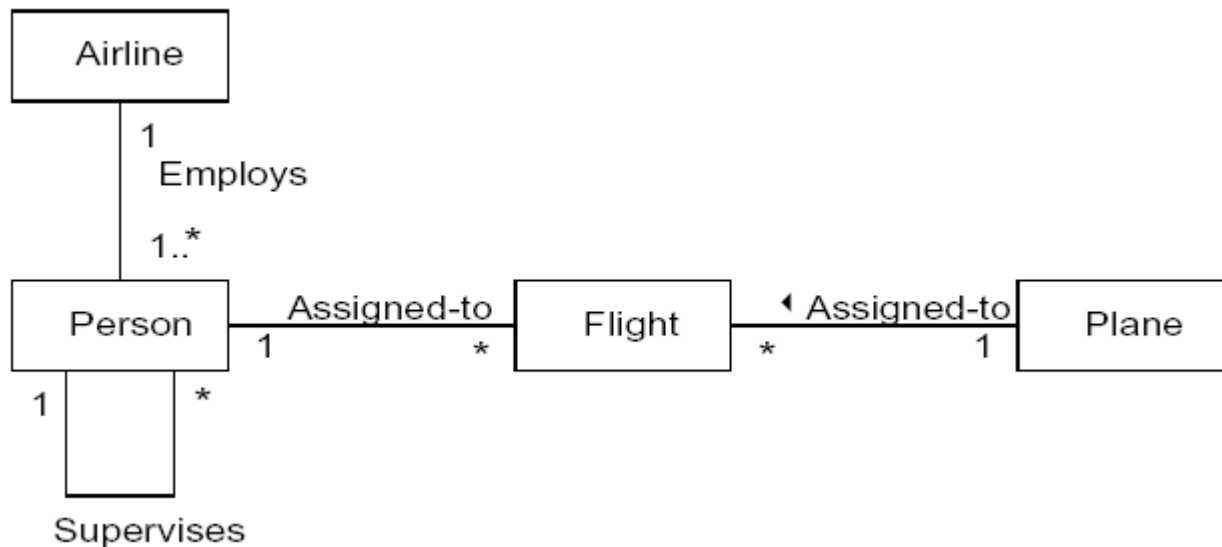
表示法

- "reading direction arrow"
- it has **no** meaning except to indicate direction of reading the association label
- often excluded

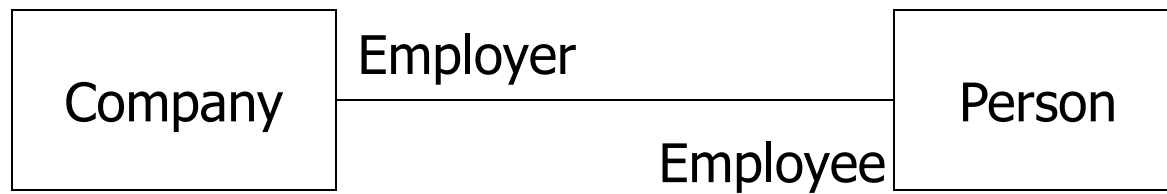


命名关联

- 使用 *ClassName-VerbPhrase-ClassName* 格式
 - 名词部分是关联的名字
- 例: *Flight – Flies-to - City*



角色

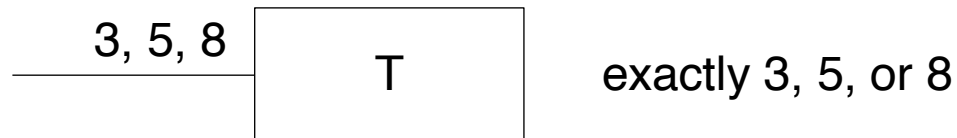
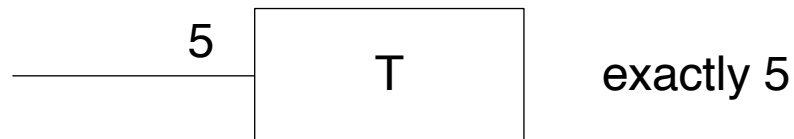
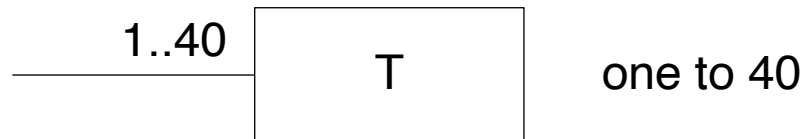
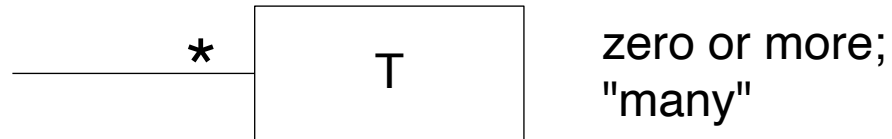


多重性

- 定义一个**A**的实例能够和多少个**B**的实例关联。
- 例子：
 - 店铺里有很多商品



多重性



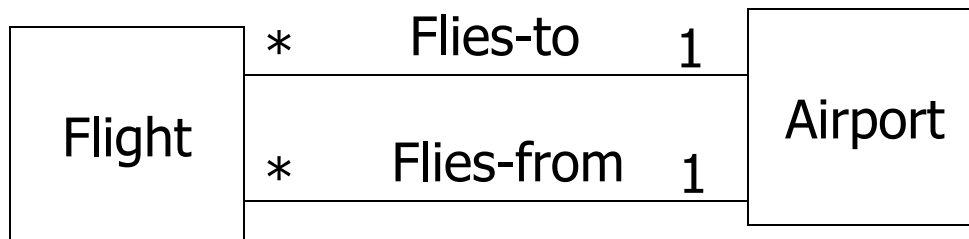
多重性

□ 多重性的语境相关



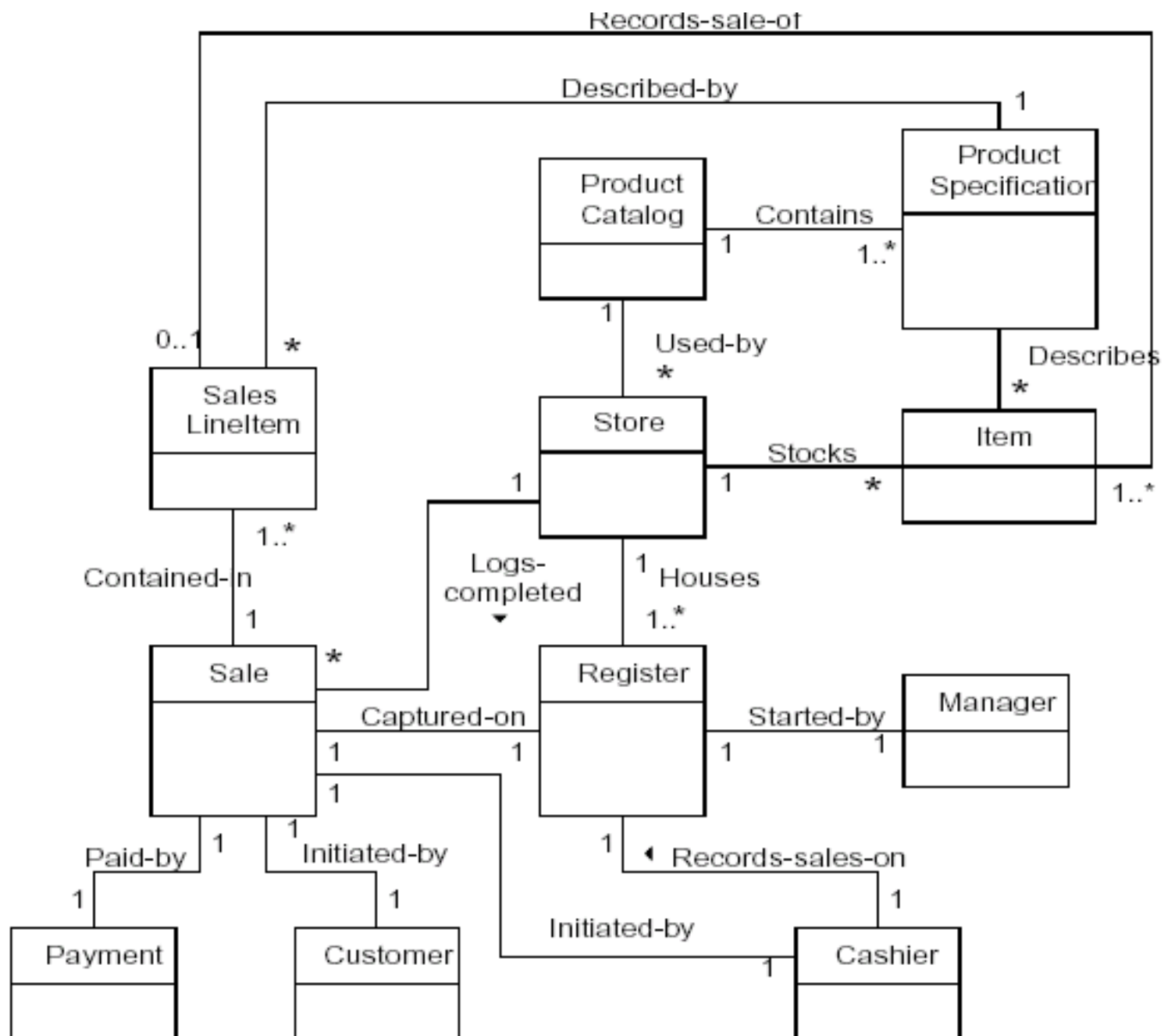
两个类之间的多重关联

- *Flight – Flies-to – Airport*
- *Flight – Flies-from – Airport*



两个类之间的多重关联



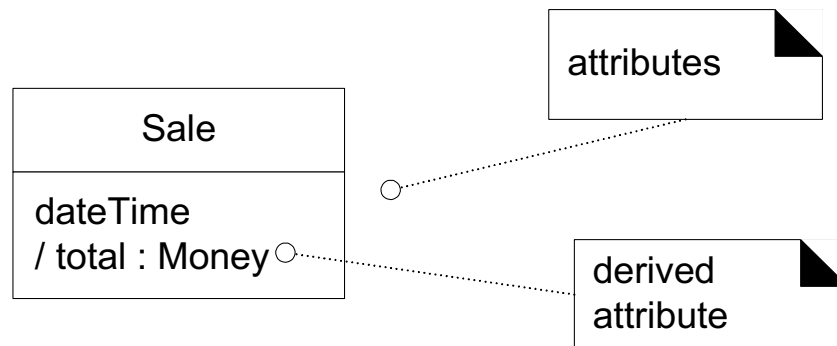
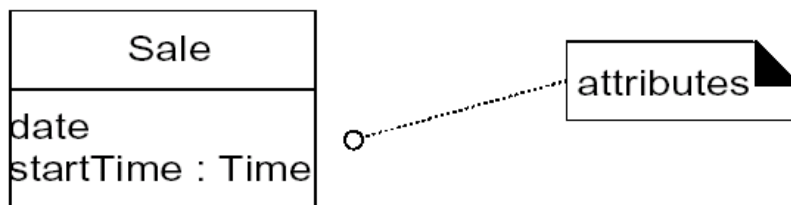


什么是属性

□ 属性是对象的逻辑数据。

□ 属性的表示法

□ 导出属性

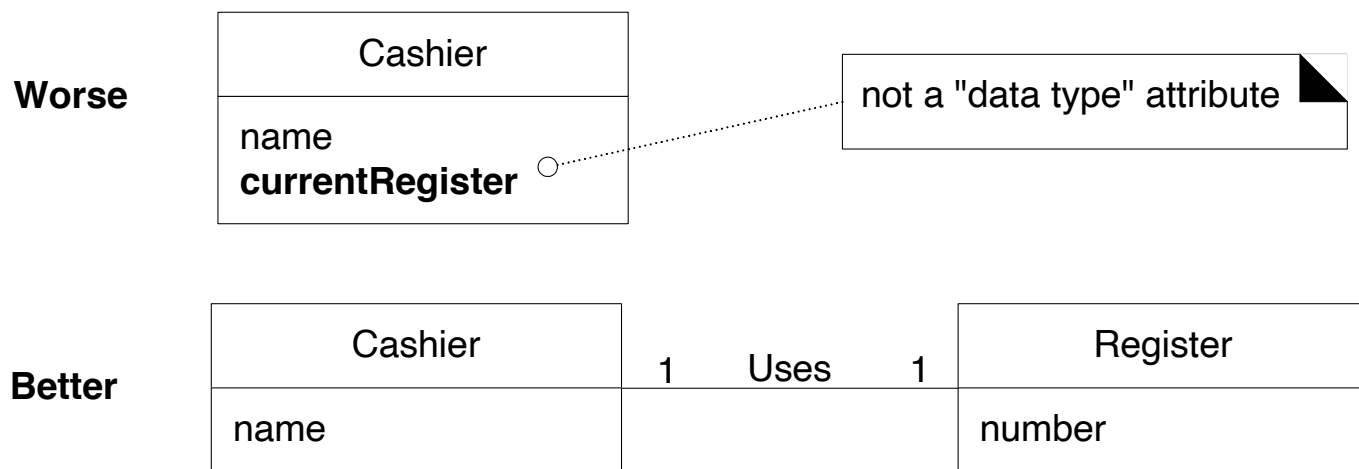


恰当的属性

- 不要将概念处理为属性
- 不要将复杂属性处理为简单属性

恰当的属性

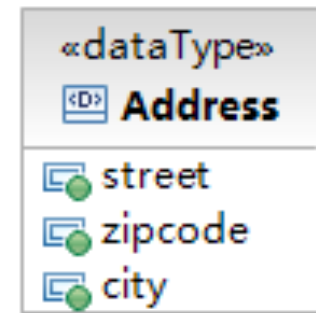
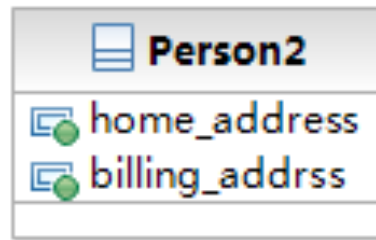
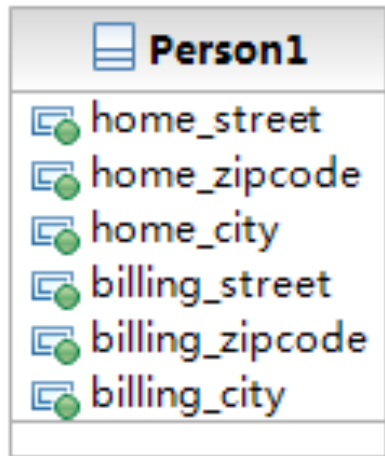
- ❑ 领域模型中的属性应该是基本数据类型(Data Type), 比如 **Boolean**, **String** 等（值对象）。
- ❑ 数据类型和概念类之间有什么区别？
- ❑ 通过关联而不是属性表示概念之间的关系。



扩充基本数据类型

- ItemID仅仅是一个Number吗？还是应该定义一个ItemIdentifier类型？
- 何时定义新的数据类型？
 - 准则：
 - 可以分割成不同的部分。
 - 电话号码, 人名, 地址
 - 具有与之相关的操作，例如解析或校验
 - 日期

示例：对地址进行建模



对数量和单位建模

- Analysis Patterns - Reusable Object Models(Martin Flower)
- 大多数数值类型都不应该简单的用Number来表示，因为需要表示单位.
- An example: Money

对数量和单位建模

Person

height : Number
weight : Number
blood glucose level : Number

worse

Person

height : Quantity
weight : Quantity
blood glucose level : Quantity

Quantity

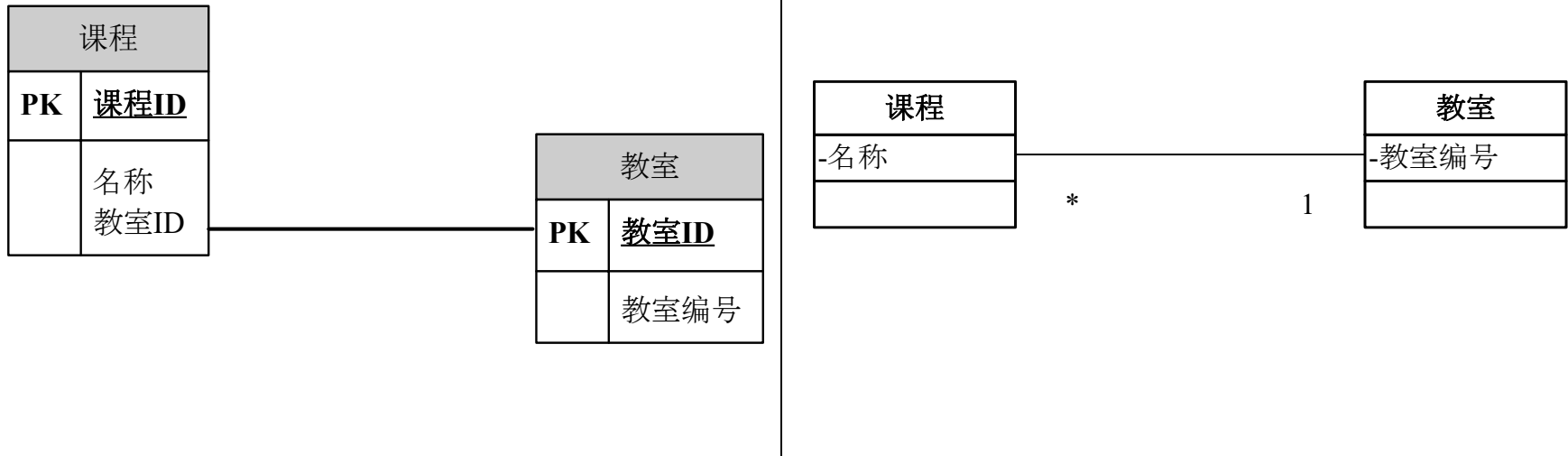
amount : Number
units : Unit
+, -, *, /, =, >, <

better

任何属性都不表示外键

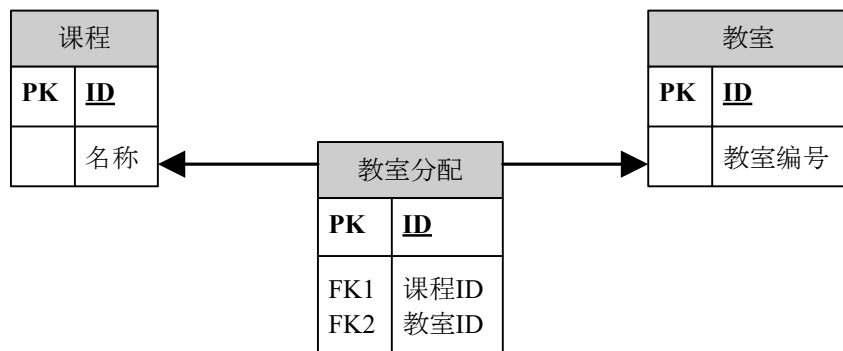
- ❑ 属性不应该用来表示概念类之间的关系。
- ❑ 违反这一原则的常见情况时像在关系数据库设计中那样增加一种外键属性(**foreign key attribute**)。外键是实现对象间关系的一种方法。
- ❑ 概念模型和ER图的关系

ER Diagram and Class Diagram

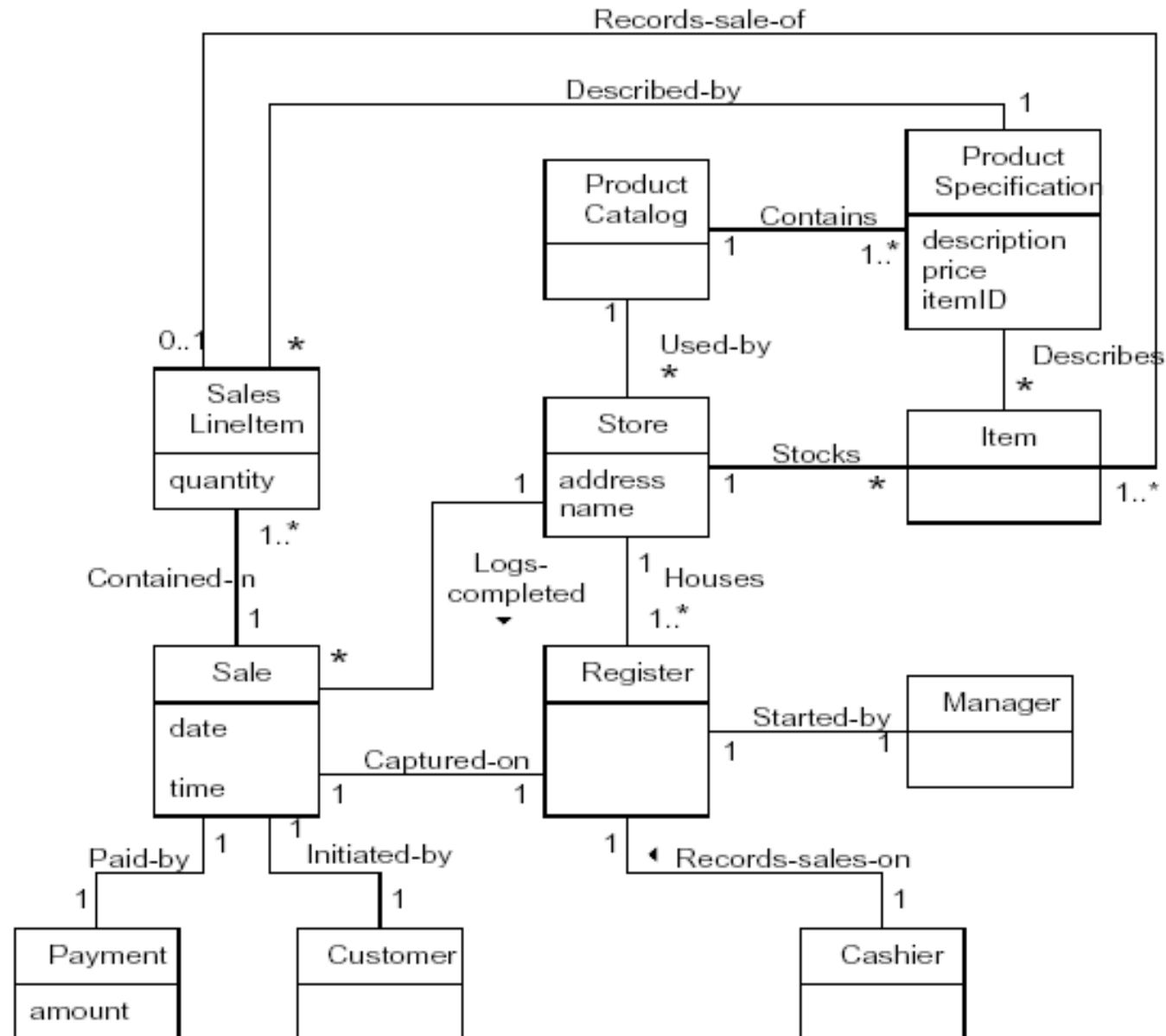


多对多的情况？

多对多的情况



Domain Model of POS System



小结

- 什么是领域模型
- 识别领域概念
- 建立概念间的关联
- 识别概念的属性

领域模型补充



内容

- 几个领域模型
- 分析模式简介

例：出版物收藏个人信息系统

□ 目标

- 管理自己收藏和阅读过的各种出版物
- 收集和记录有关出版物的阅读信息
 - 他人的评论
 - 自己的读后感

出版物收藏个人信息系统

□ 出版物类型

■ 书

- 纸质

- 电子书

■ DVD (可选)

- DVD介质

- 计算机文件

□ 评论

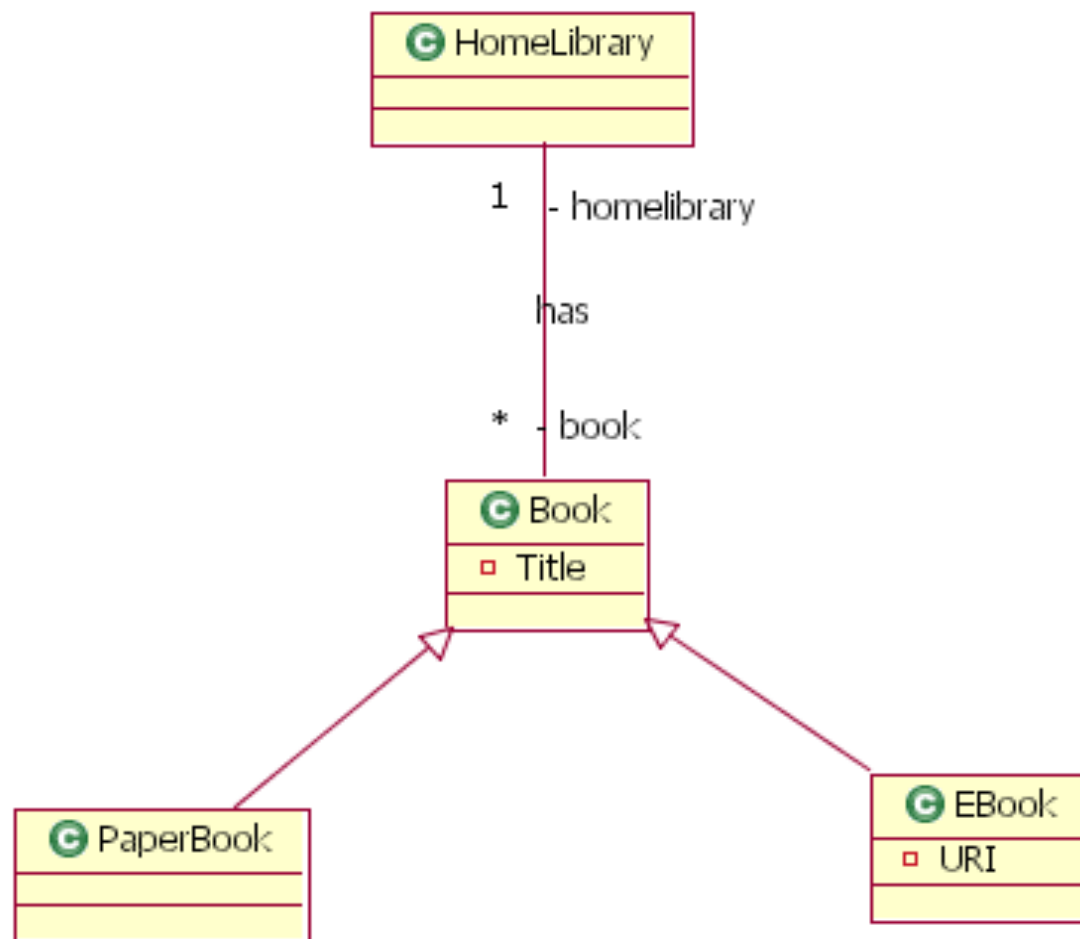
- 个人评论

- 他人评论

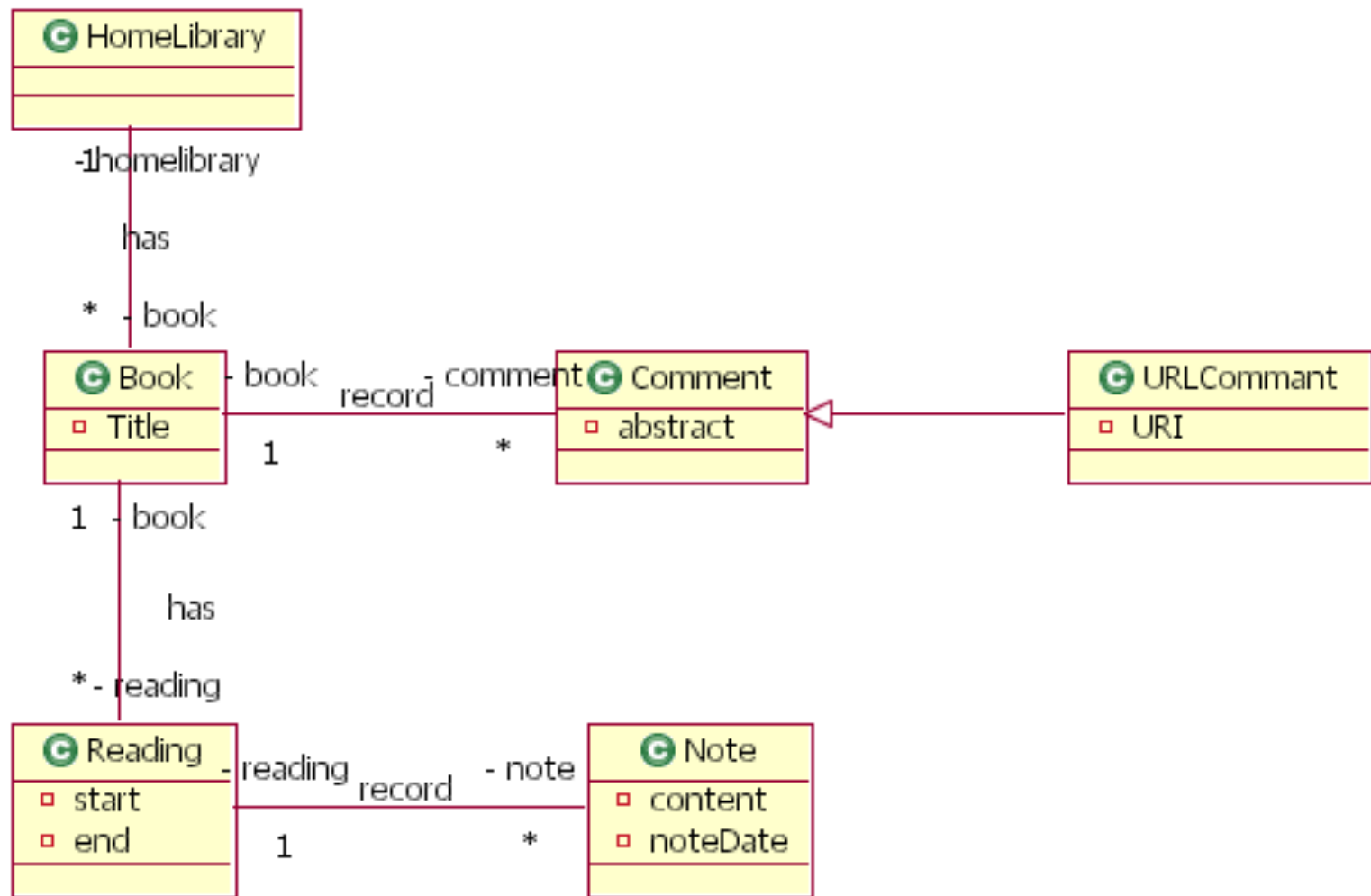
用例：阅读书籍

- 选择一本书开始阅读，告诉系统这本书处于阅读中。
- 写读书日志（可能有多篇），并录入系统。
- 可以从网络或其他杂志收集该书的书评，记录到系统中。
- 阅读完毕后，系统记录本次阅读完毕。

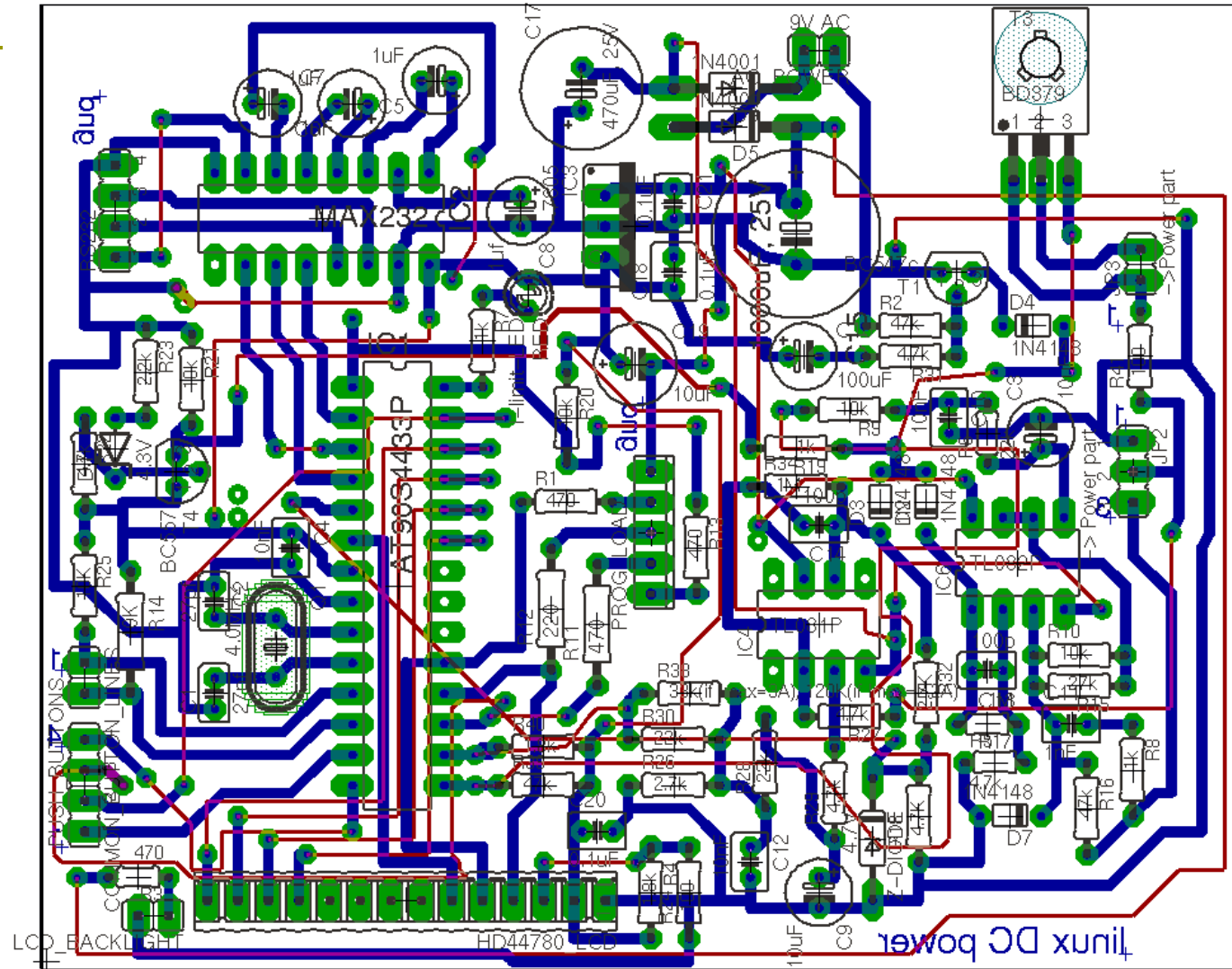
书籍：核心



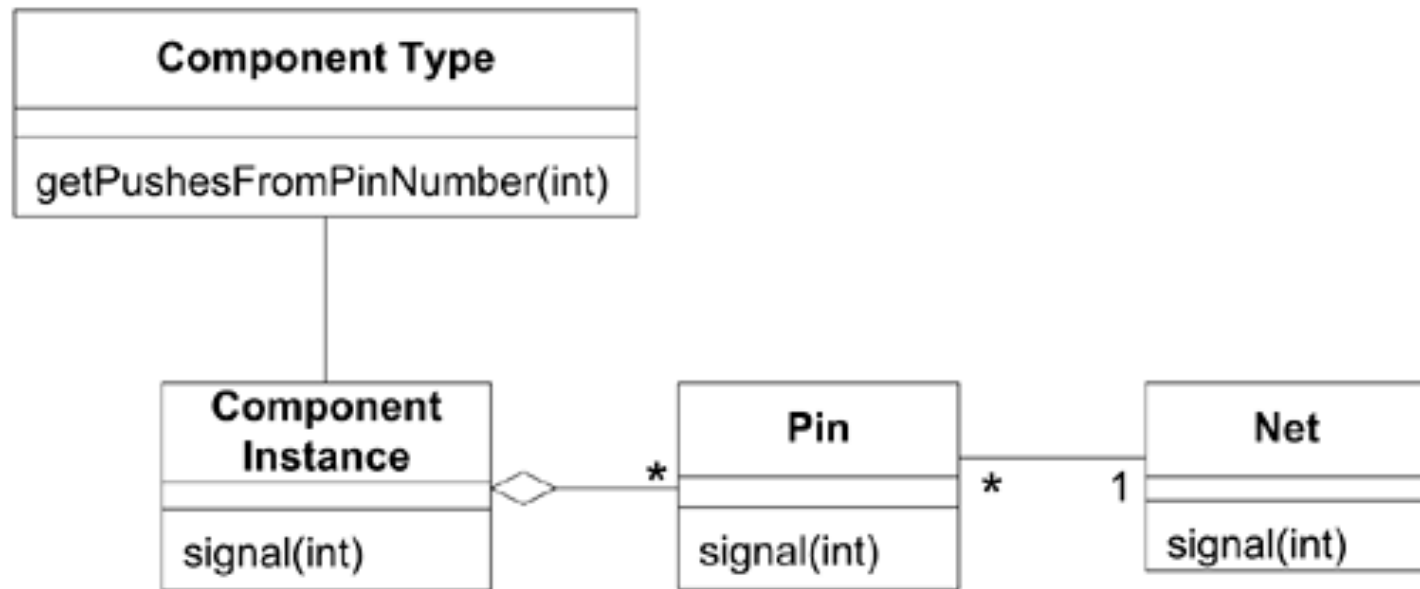
与阅读相关的领域模型



一个电路模型



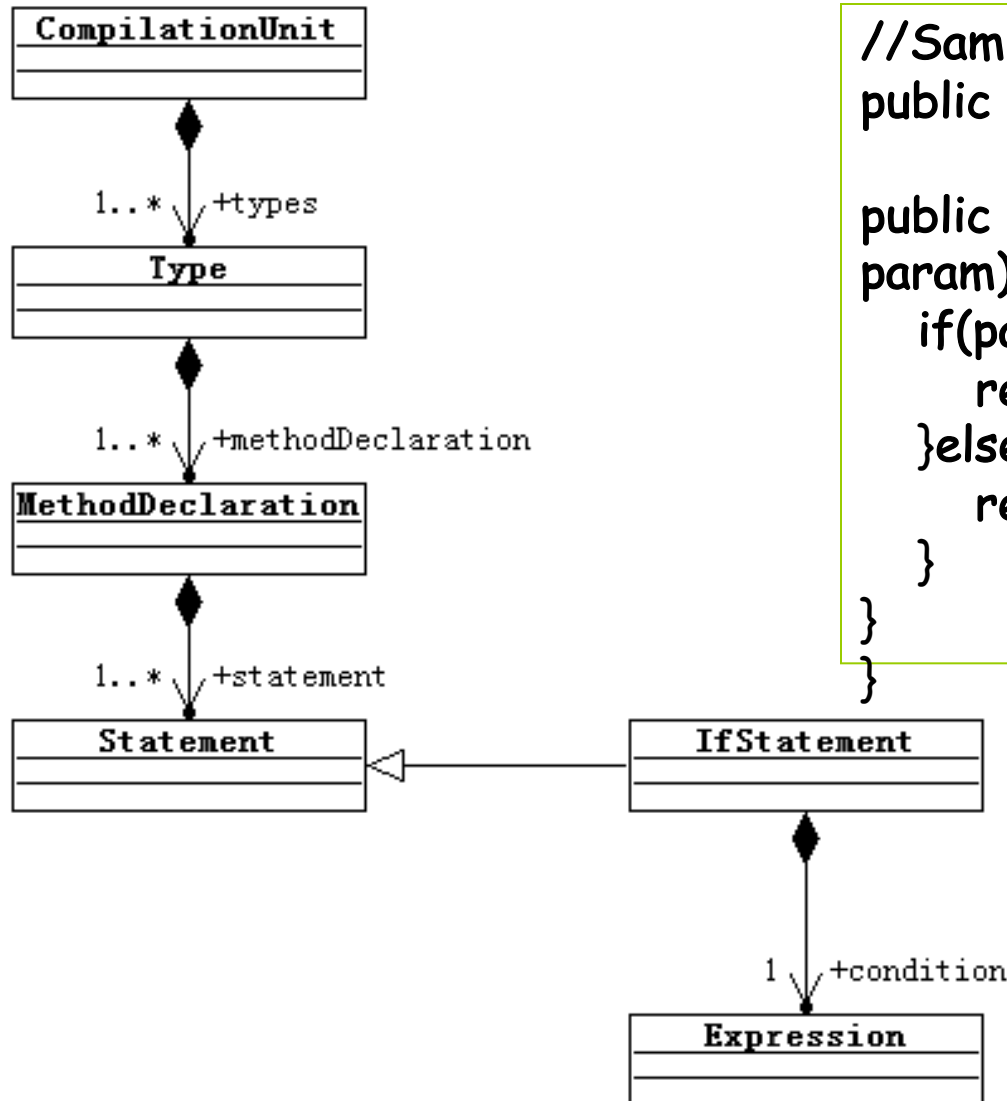
一个电路模型



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

By [Eric Evans](#)

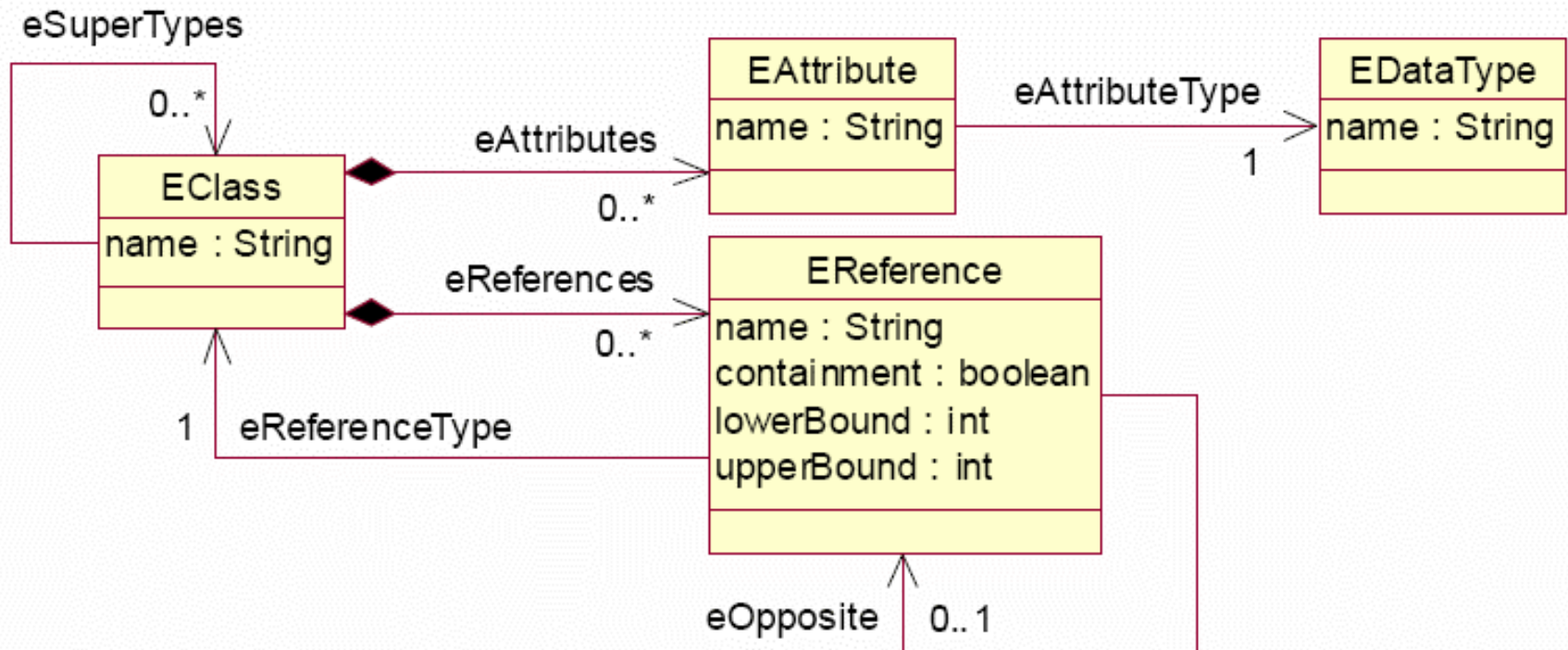
Java的抽象语法树



```
//SampleJava.java
public class SampleJava {

    public int testMethod(int
    param){
        if(param<0){
            return -1;
        }else{
            return 1;
        }
    }
}
```

UML的元模型（类）



分析模式简介

□ 模式的定义：

- 在特定的情况下，针对一个普遍问题的解决方案
[Christopher Alexander, architect]

□ 分析模式：

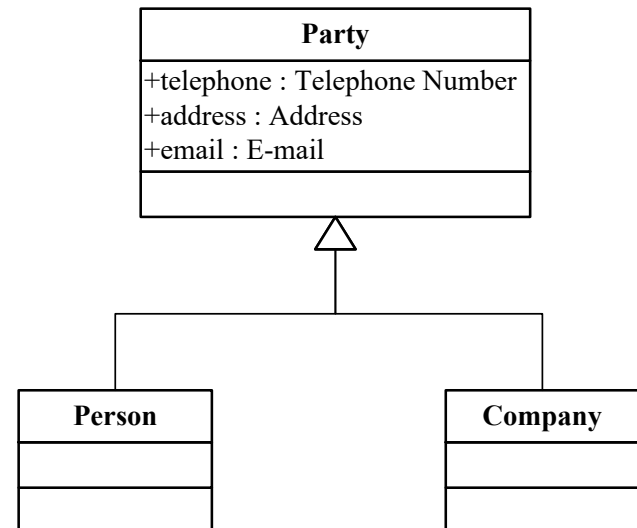
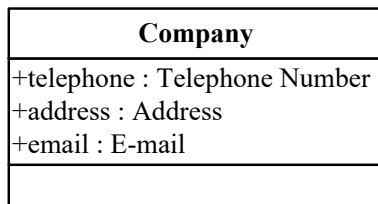
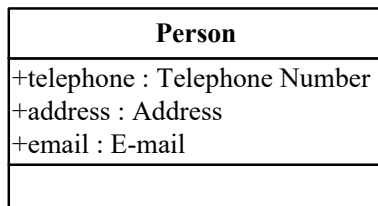
- Analysis Patterns: Reusable Object Models
Martin Fowler

□ 分析模式和设计模式的区别

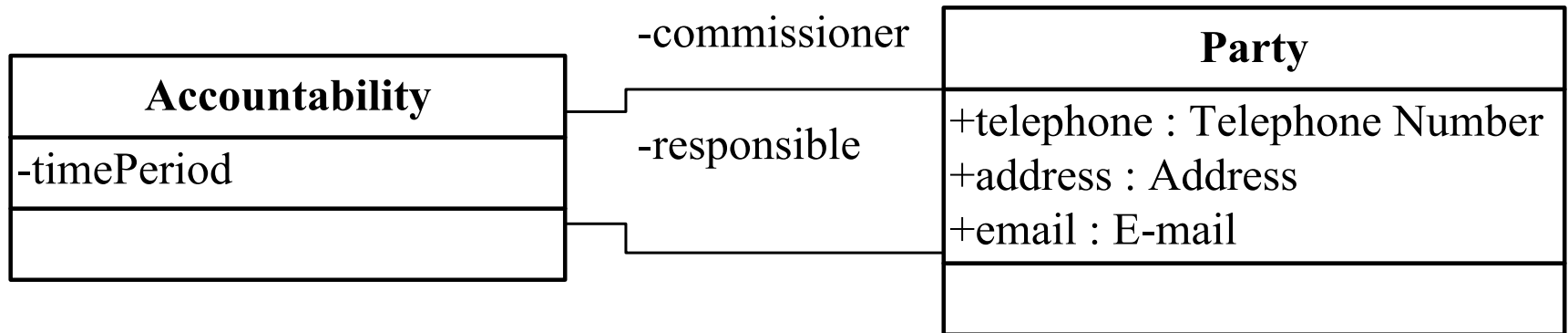
- 分析模式解决领域问题，设计模式解决设计(技术领域)问题。

Party(当事人)

我们经常需要和人与组织打交道，如何建立这个领域的概念模型？



Accountability(责任关系)



1. 复旦大学(**Company Party**)和我(**Person Party**)建立了一种雇用责任关系(**Accountability**)，其中复旦大学是委托方(**commissioner**)，我是责任方(**responsible**)
2. 教研组和**ABC**公司签订了软件开发合同(**Accountability**)，其中委托方(**commissioner**)为**ABC**公司，责任方是教研组(**responsible**)

Account/Transaction(帐户 / 交易)

Account
-balance

Account
-balance

Entry
-ammount

1

*

Account
-balance

Entry
-ammount

1

*

Transaction
-datetime
-operator

2

1

Account/Transaction(帐户/交易)

- 使用信用卡支付超市购买商品的费用**500元**
 - 两个账户(个人账户和商场账户)
 - 两笔明细(个人账户明细和商场账户明细, 各为-**500元**和**500元**)
 - 一次交易
- 生产车间从公共仓库领取**5桶油漆**
 - 两个账户(工厂现场仓库账户和公共仓库账户)
 - 两笔明细(工厂现场仓库账户明细和公共仓库账户明细, 各为-**5桶**和**5桶**)
 - 一次交易

其它模式

- 测量、观测(Observations and Measurements)
- 计划(planning)
- 贸易(trading)

练习

- 为一个网上书城的部分设计一个模型
 - 与图书关联的信息有：标题，作者，出版社(名称，地址)，价格，分类，用户评论列表。
 - 对于每个用户的评论，需记录用户**ID**、评论内容和星级。