
Rottingen

面向对象分析 （Object-Oriented Analysis, OOA)

- 面向对象分析技术关注应用领域中的实体，并将其建模为对象
- 面向对象分析技术主要基于分类、泛化、聚合关系在对象集合之间建立结构
- 对象的行为是执行预定的动作（服务/活动）
- 对象通过执行动作来完成状态变迁

面向对象分析的起源

- 面向对象程序设计（OOP）[Booch86]

  将OOP中的概念上推到分析和设计阶段

- 数据库设计（Database design） [Chen 76]

  将数据语义建模概念，如实体-关系、泛化、聚合、和分类用于系统分析和设计

- 结构化分析（Structured Analysis ） [Ross 77]

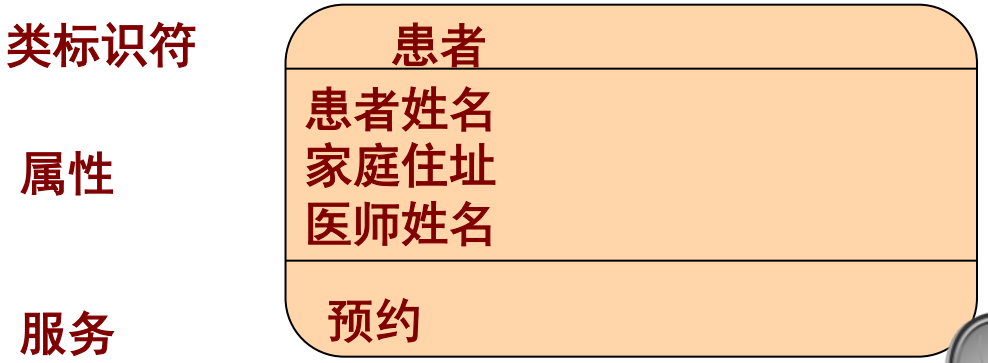
  将结构化分析方法与技术，如SADT方法等用于系统分析与建模

- 知识表示（Knowledge Representation） [Borgida85]

  采用基于问题框架和语义网络的知识表示方法

面向对象分析方法举例

- Peter Coad的面向对象方法［Coad91］
- “对象”是问题领域中真实存在的实体，有“定义清晰的边界”
- 对象中封装有属性和行为
- 面向对象分析的五个核心概念：
  - 对象、属性、结构、服务和主题



软件系统建模原则

- 支持修改和重用
  - **有经验的工程师重用已有的设计**
    - 模型组件重用
    - 模型结构重用
  - **有智慧的工程师规划未来**
    - 建立可重用的模型组件
    - 建立易于修改的模型结构

- 五大有力武器：
  - **抽象：**关注重点，暂忘细节
  - **分解：**将问题切分，分而治之
  - **多视角映射：**切换关注点，分别讨论
  - **模块化：**

    建立稳定结构、缩小变化影响范围
  - **模式：**多次成功应用的模型结构

对象建模原则1: 抽象

Source: *Adapted from Davis, 1990, p48 and Loucopoulos & Karakostas, 1995, p78*

- 抽象
  - **在对象间找出共性，忽略不相关细节**
  - **关注对象间的一般/特殊关系**
    - 将具有相同属性或角色的对象放入同一个类集合中
    - 再通过父子关系，将由共性的类定义为同一个父类的子类
- 例如:
  - 需求是处理航天器故障
  - 将故障按不同的故障类型纵向分类。

based on location:

- instrumentation fault,
- communication fault,
- processor fault,
- etc

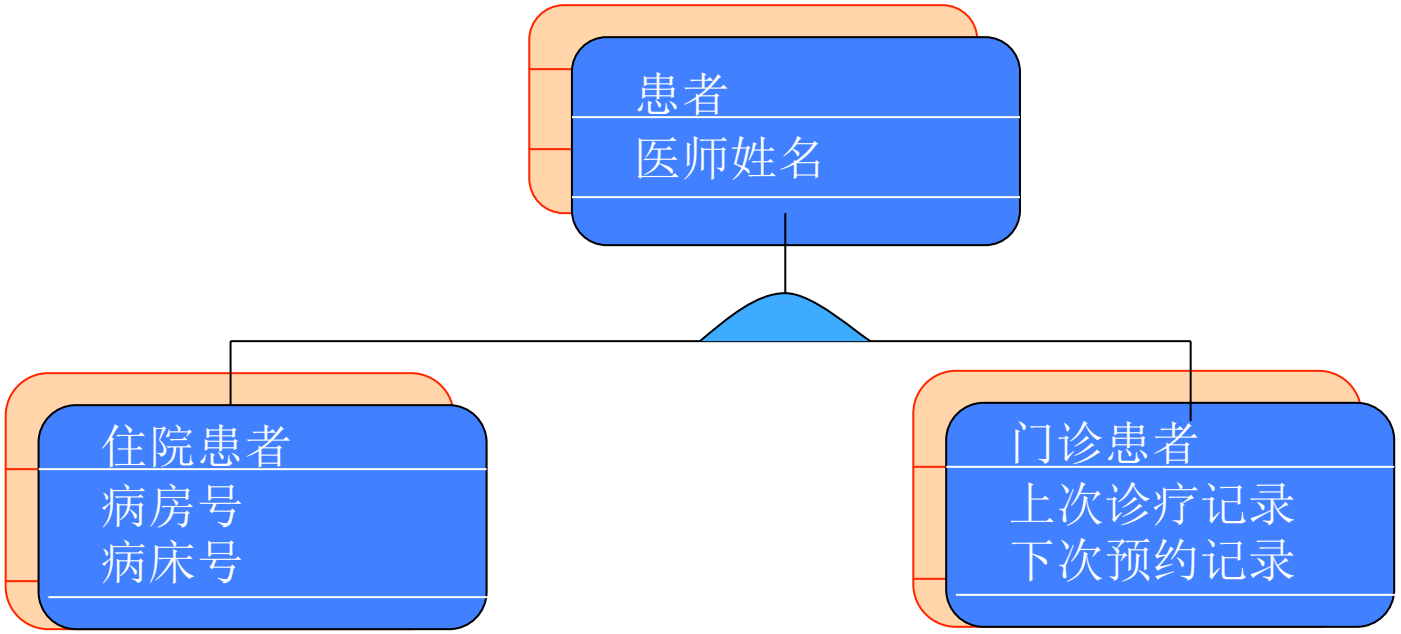
OR

based on symptoms:

- no response from device;
- incorrect response;
- self-test failure;
- etc...

继承/一般-特殊结构（Gen-Spec Structures）

- 一般-特殊结构将类组织成基于继承关系的分类层次结构
- 自底向上是从特殊到一般的类 (generalization）
- 自顶向下是从一般到特殊的类（specialization).



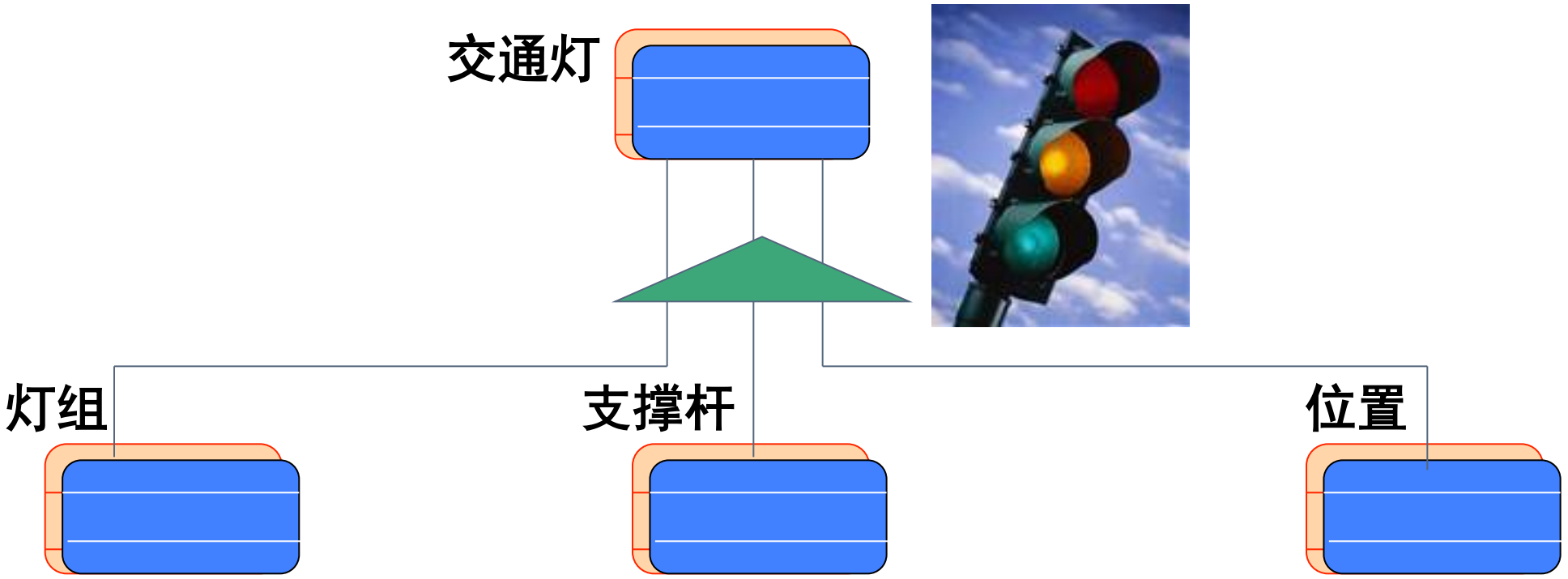
对象建模原则2：分解

- 分解
  - 表达整体部分关系，细分为聚合和组合
- 例如：
  - 目标是飞行器研发
  - 将问题分解为子系统研发：
    - 导航系统；
    - 数据处理系统；
    - 指挥控制系统；
    - 环境控制系统；
    - 等

  这是我们对问题的分解
    - 现实世界中的设计可以组件化
    - 系统分解方式决定系统的体系结构设计

整体-部分结构（Whole-Part Structures）

- 整体部分结构描述对象间的组合关系. 例如，一个交通灯对象由0-3个灯组，支撑杆和位置对象组合而成。

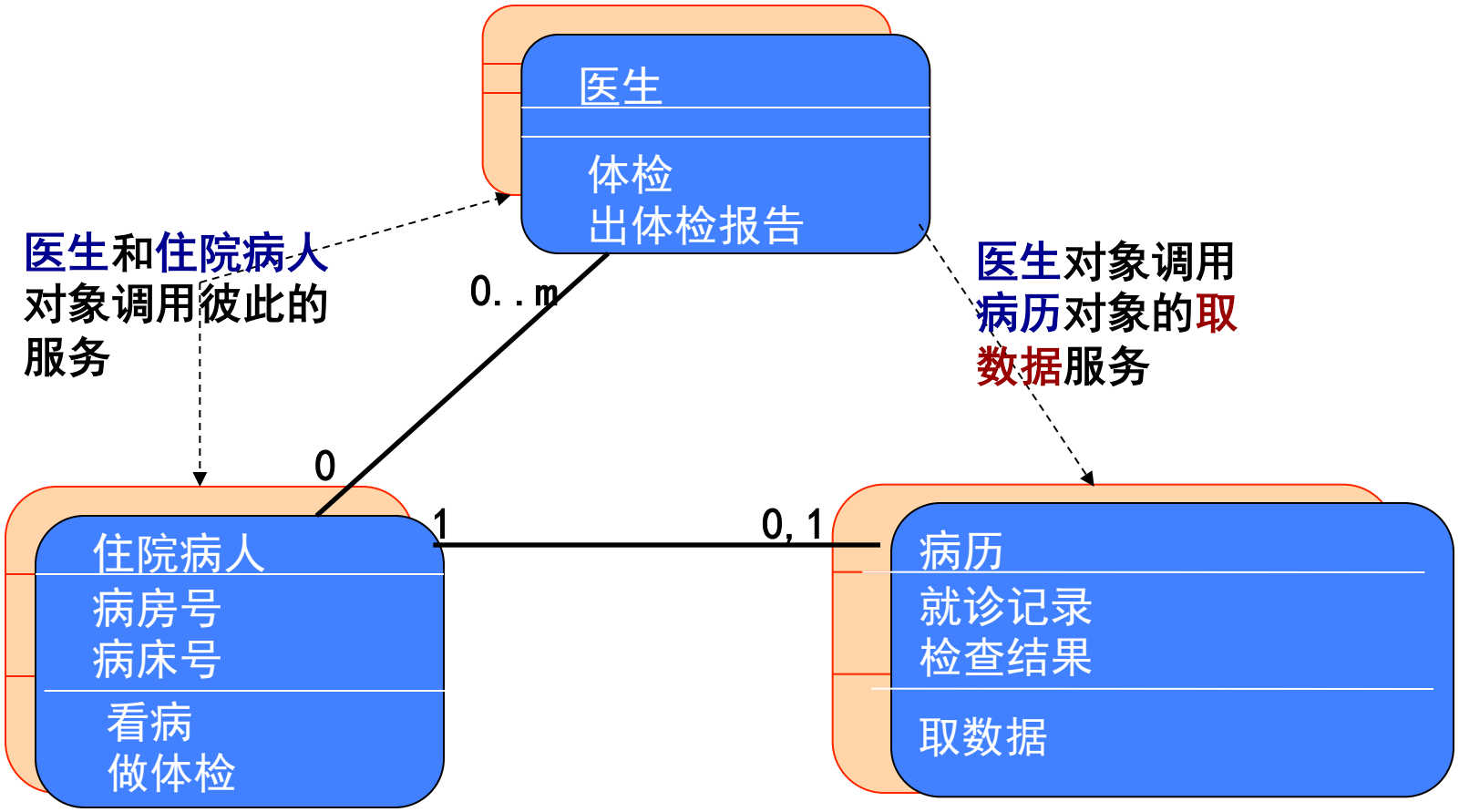


服务建模（Services）

- 对象为其周遭的其他对象提供服务，例如，医生对象对外提供的服务包括：体检，出体检报告等。
- Coad 的OOA方法中，定义了三种类型的服务：
  - 瞬时服务（Occurrence services）：对象的创建、结束，修改等等
  - 计算服务（Calculate services）：对象为其他对象完成计算任务等
  - 监控服务（Monitor services）：对象持续监控流程，检查预设条件是否满足
- 我们用带箭头的虚线来表示一个对象引用另一个对象的服务

从面向对象到面向服务，是看待问题的视角的切换

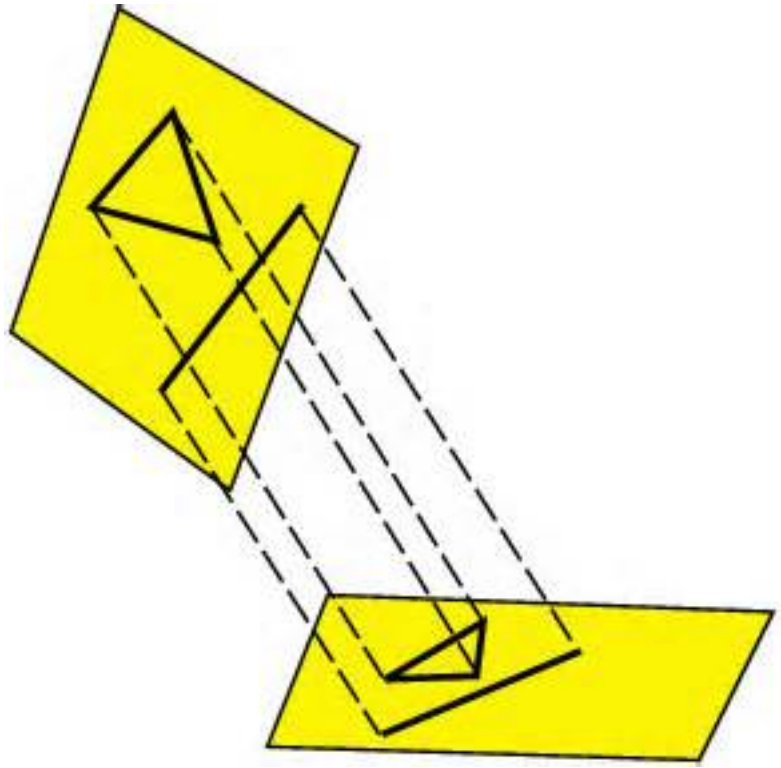
服务关系 （Services relationships）



对象建模原则3：投影

Source: *Adapted from Davis, 1990, p48-51*

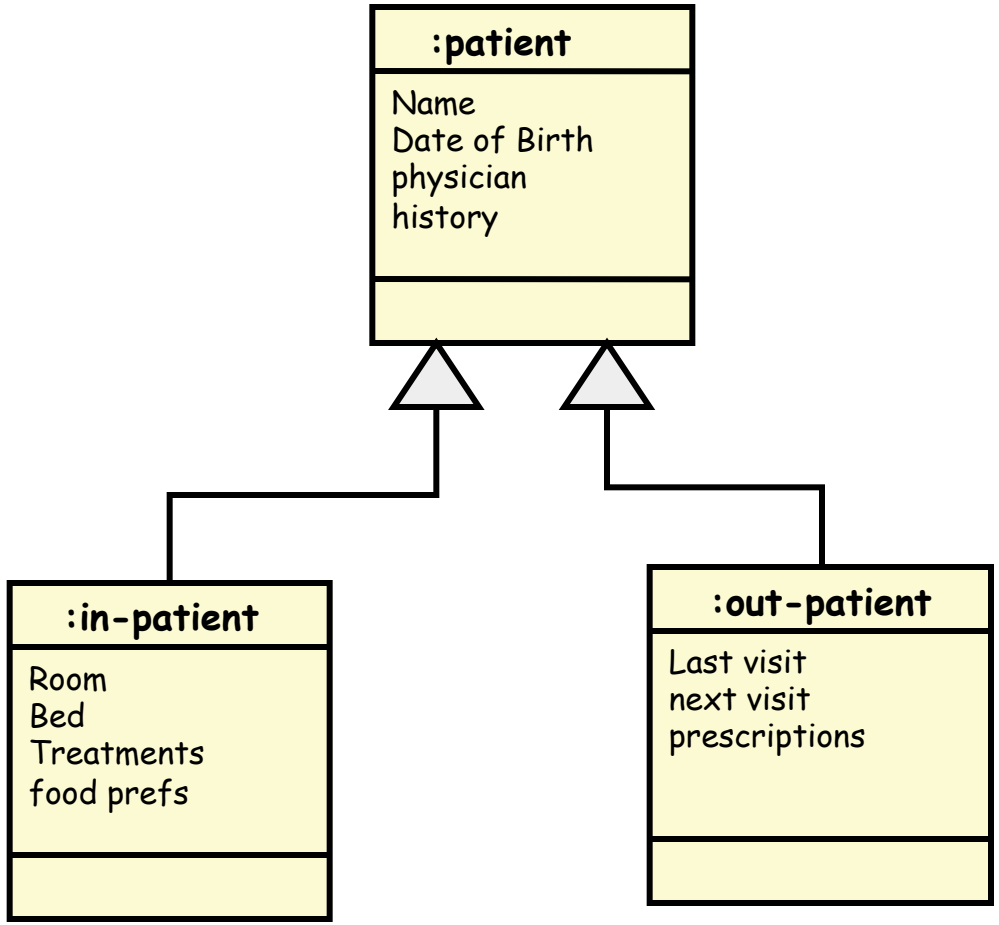
- 投影：
  - **从多个视角分别建模问题的不同方面**
    - 一如建筑施工中的不同视角的图纸
- 例如：
  - **需要进行飞行器需求建模**
  - **投影建议分别建模：**
    - 安全性、容错性、实时性 …
- 注意：
  - **投影和分解有共同点：**
    - 分解定义整体-部分关系
    - 投影定义视图
    - **分解的假设是子模块间依赖性较小**



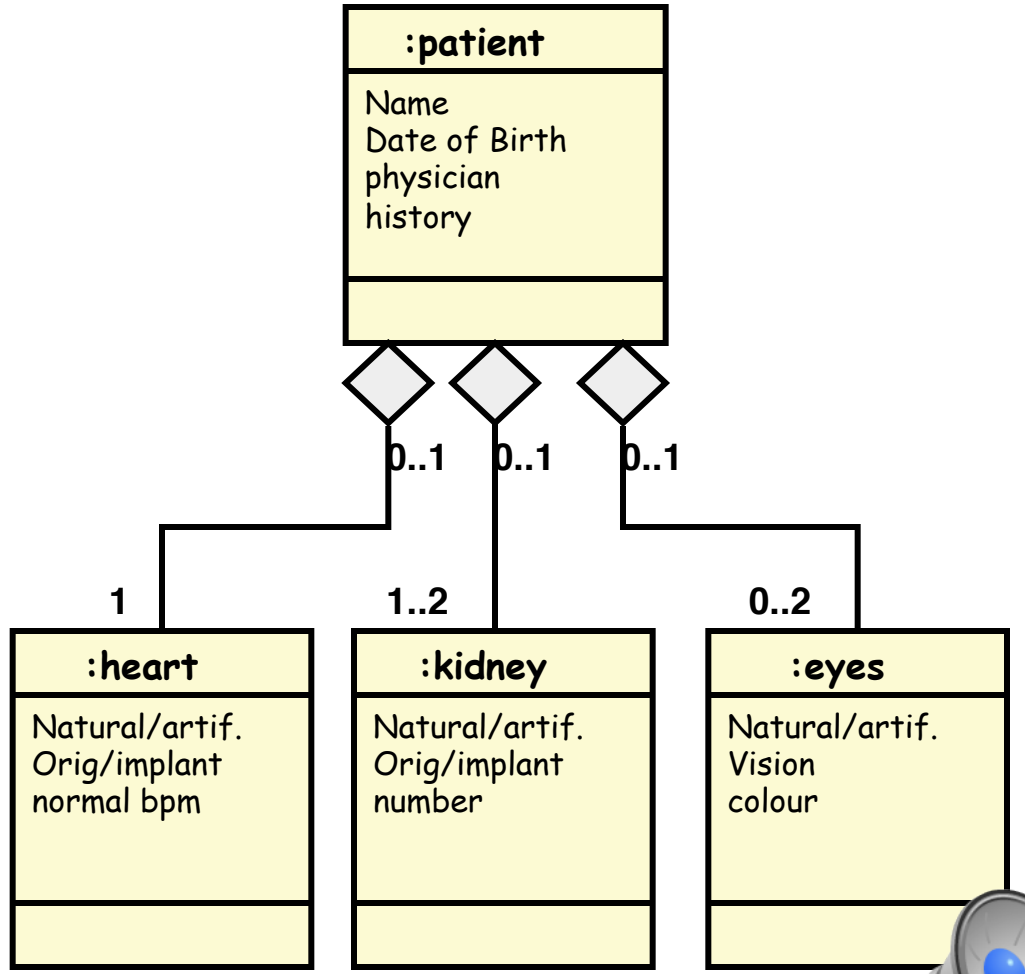
UML类图模型例

Source: *Adapted from Davis, 1990, p67-68*

继承 Generalization (an abstraction hierarchy)



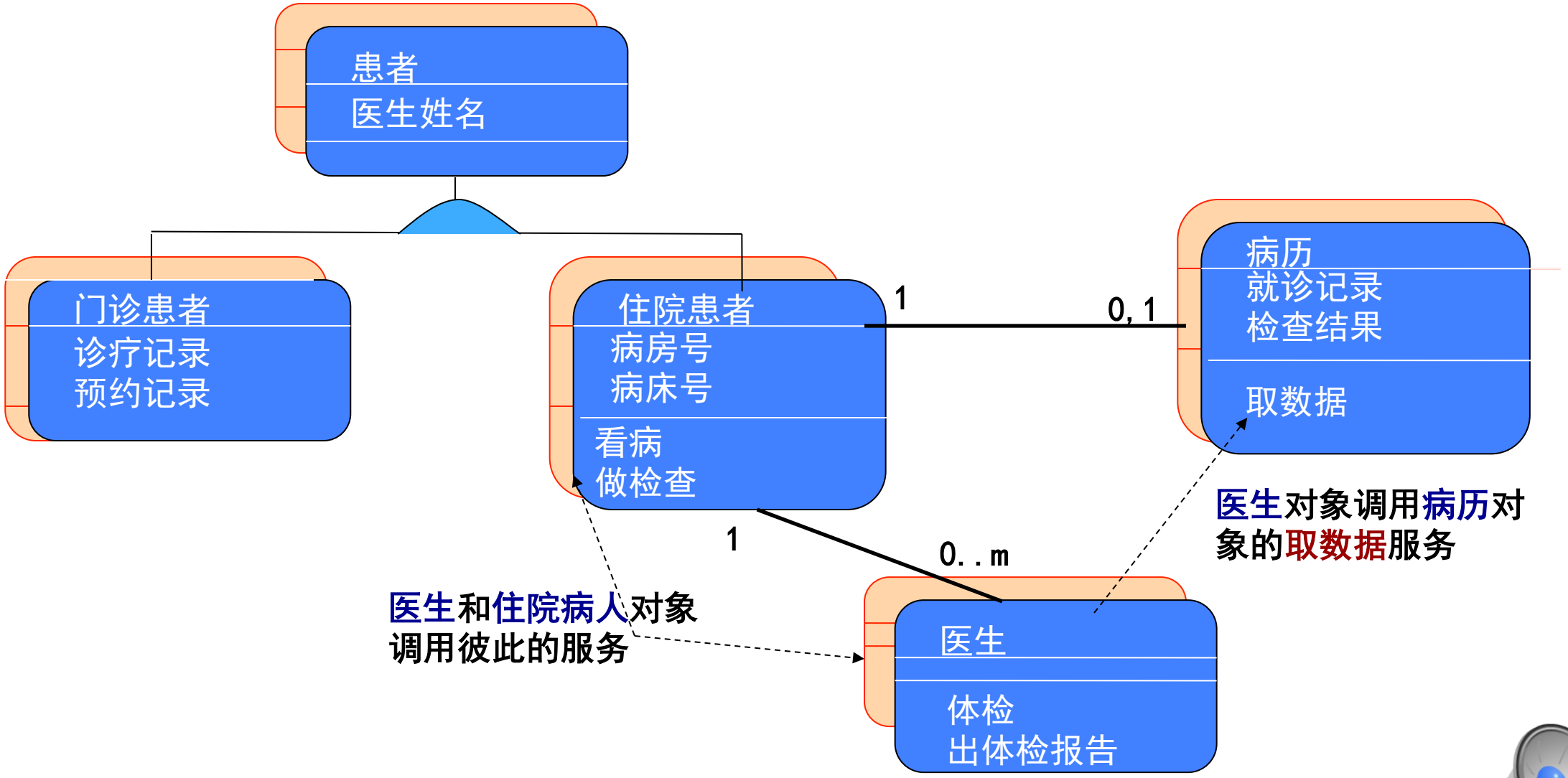
组合Aggregation (a partitioning hierarchy)



面向对象的分析方法学

- 识别对象和类（类是对象的抽象定义）
- 识别类之间的关系，建立由继承和组合关系组成的类层次结构
- 定义主题，通过主题将对象模型组织成多个抽象层次或视角，一般说来通过继承关系或整体部分关系联系起来的类同属于一个主题
- 识别各个对象内部的属性信息，并将其赋予相应抽象层次的类
- 为每个类定义服务

面向对象分析的局限性


患者
医生姓名
门诊患者
诊疗记录
预约记录
住院患者
病房号
病床号
看病
做检查
1
0, 1
病历
就诊记录
检查结果
取数据
1
0. . m
医生
体检
出体检报告
医生对象调用病历对象的取数据服务
医生和住院病人对象调用彼此的服务

面向对象分析常用术语对照表

OOA	OOSE (Jacobson)	OOD (Booch)	OMT (Rumbaugh)
		Metaclass	
Object	*Instance*	*Object*	*Object*
Gen-Spec	*Inheritance*	*inherits*	*Generalization*
Whole-Part	*Consists-of*		*Aggregation*
Instance conn.	*Acquaintance*		*Link*
Message	*Stimuli*	*Message*	*Event*
Message conn.	*Communication*		
Attribute	*Attribute*		*Attribute*
Service	*Operation*		*Operation*
Subject	*~View (subsystem)*		*Sheet*
(Execution thread)	*Use case*		*~Scenario*
(user)	*Actor*