

面向对象设计方法（UML）

学习面向对象设计方法应关注哪些方面

概念
表示法
模型框架
开发过程

本课程的三个主要部分

基本概念
面向对象的分析（OOA）
面向对象的设计（OOD）

第一章 面向对象方法概论

什么是面向对象

面向对象方法的基本思想和主要特点

面向对象方法为什么有利于改进软件开发

准确掌握面向对象方法的基本概念

对象 类 属性 操作 继承

封装 关联 聚合 消息 多态性

抽象 永久对象 主动对象

继承、封装等概念有什么重要意义，是否有副作用？如何克服？

第二章 不同的分析与设计方法

历史上各种典型的分析与设计方法

大致了解其主要思想和优缺点

什么是**OOA**? 什么是**OOD**?

对其内容、目标和特点有准确理解

软件系统的分析与设计面临哪些挑战?

面向对象方法的主要优点（能说出道理）

几种典型的**OO**方法

大致了解，开阔知识面

第3章 统一建模语言UML简介

概括地了解**UML**的全貌

掌握**UML**的几个重要概念

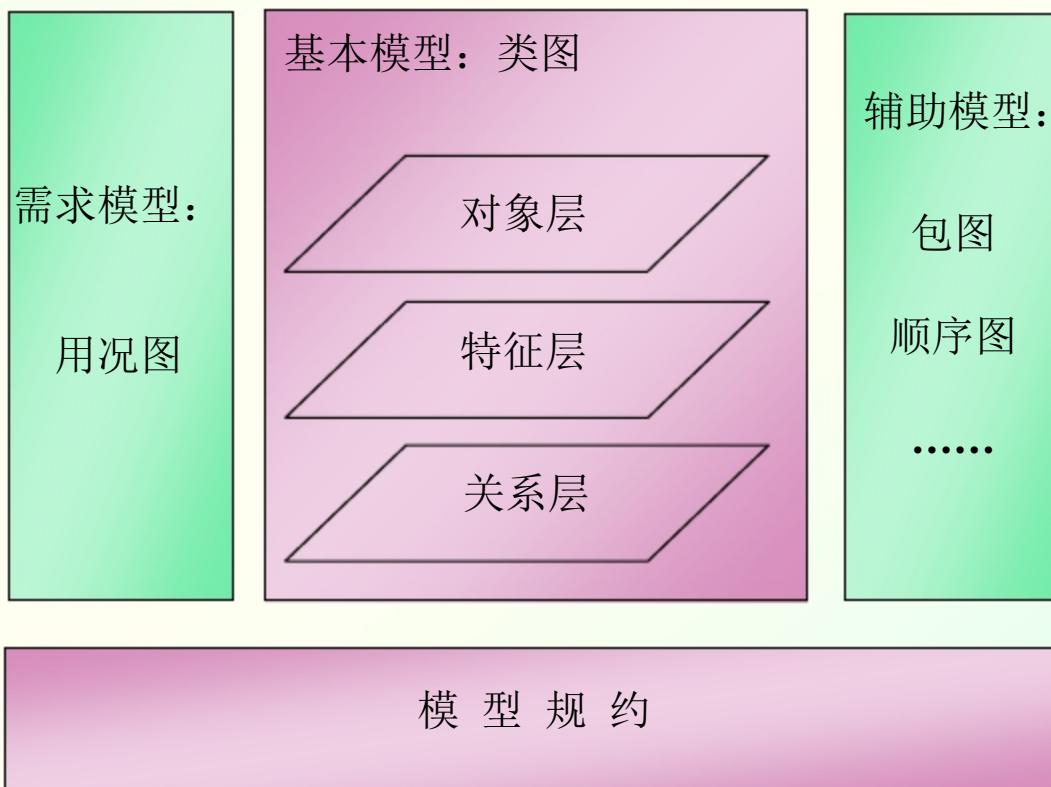
对具体内容的详细了解主要看第**5~9**章相关内容

第4章本书的OOA&D方法概貌

概念

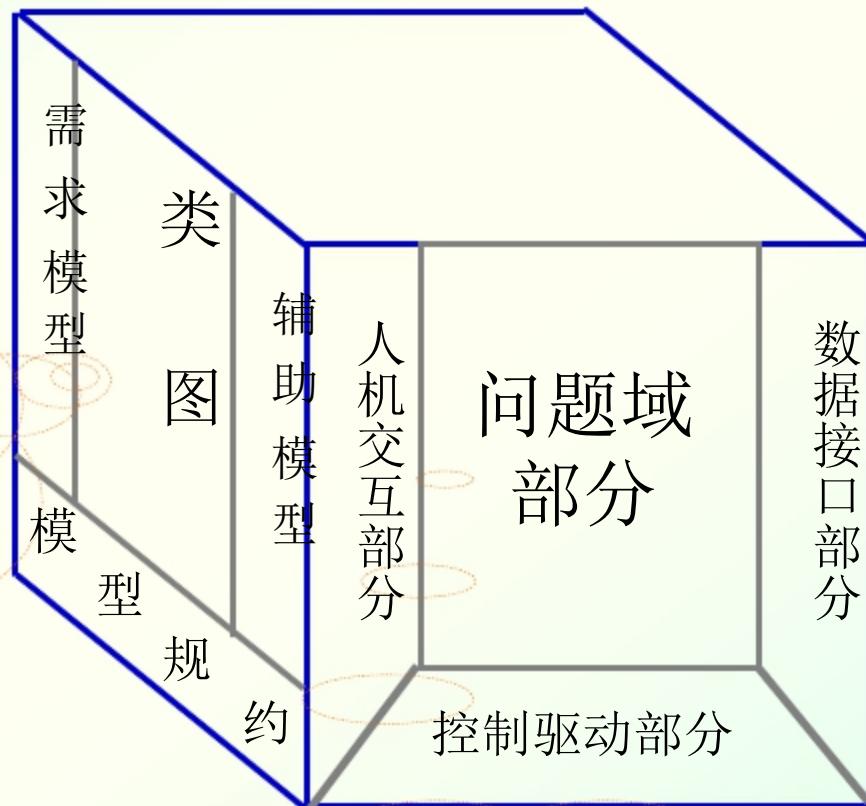
提倡充分运用**OO**方法基本概念
加强过程指导，避免过多地引入扩充概念

OOA模型框架

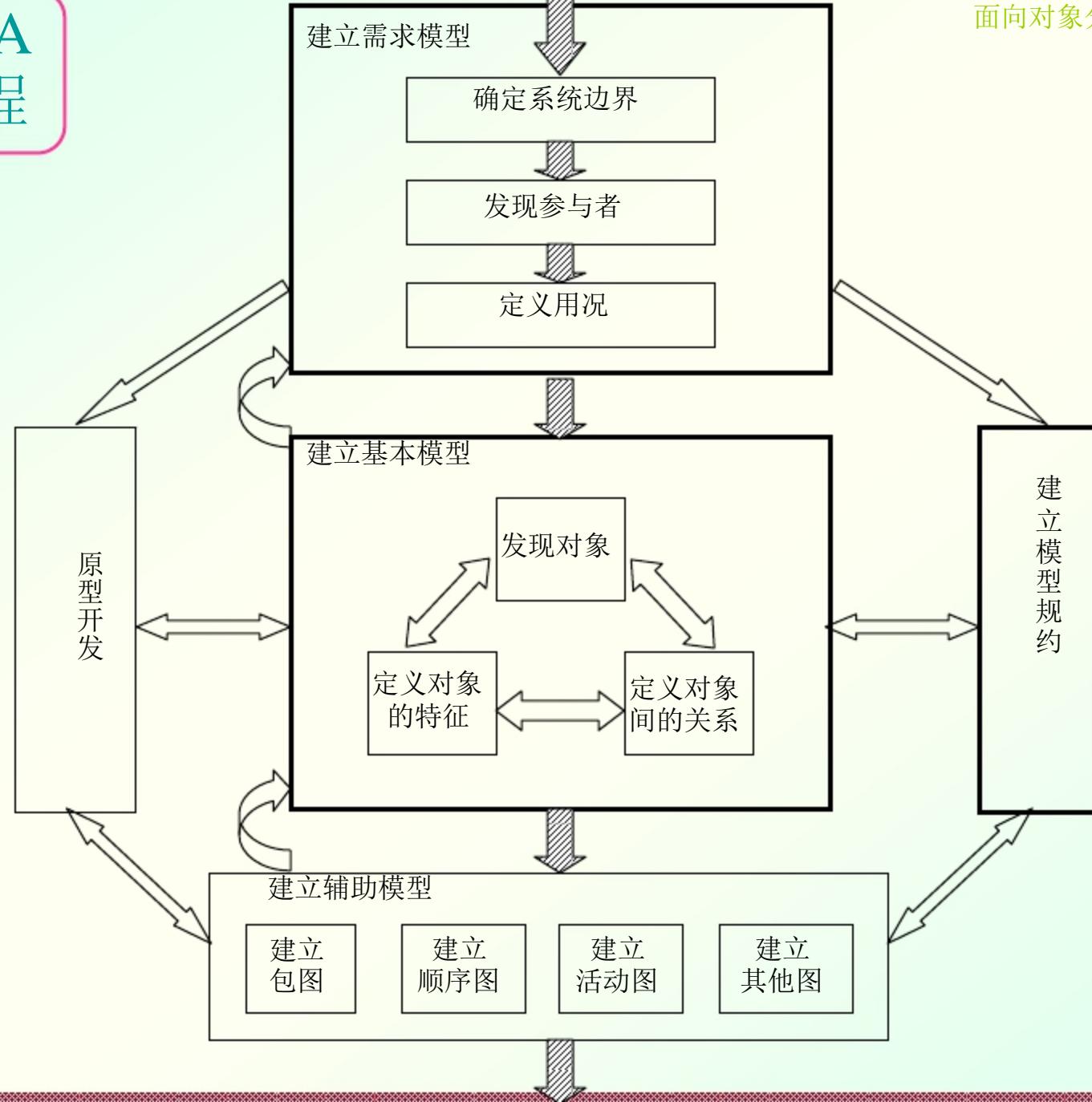


OOD模型框架

——从两个侧面来描述

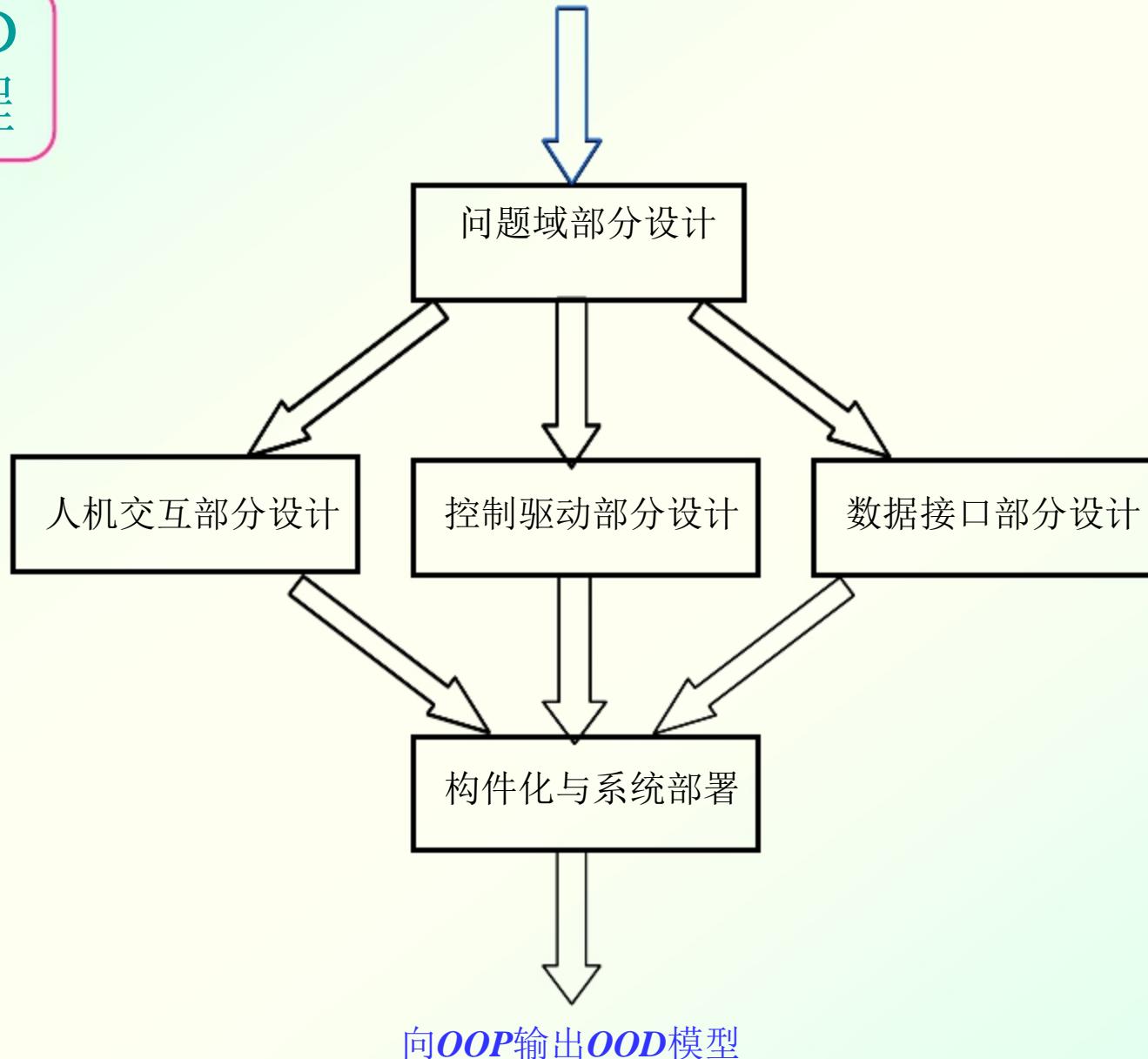


OOA 过程



输入*OOA*模型

OOD
过程



OOA与OOD的关系

一致的概念和表示法

不同的内容、目标和抽象层次

OOA与OOD两种不同的分工观点

从MDA看OOA与OOD的关系

第5章 建立需求模型

什么是用况，如何用它来描述需求

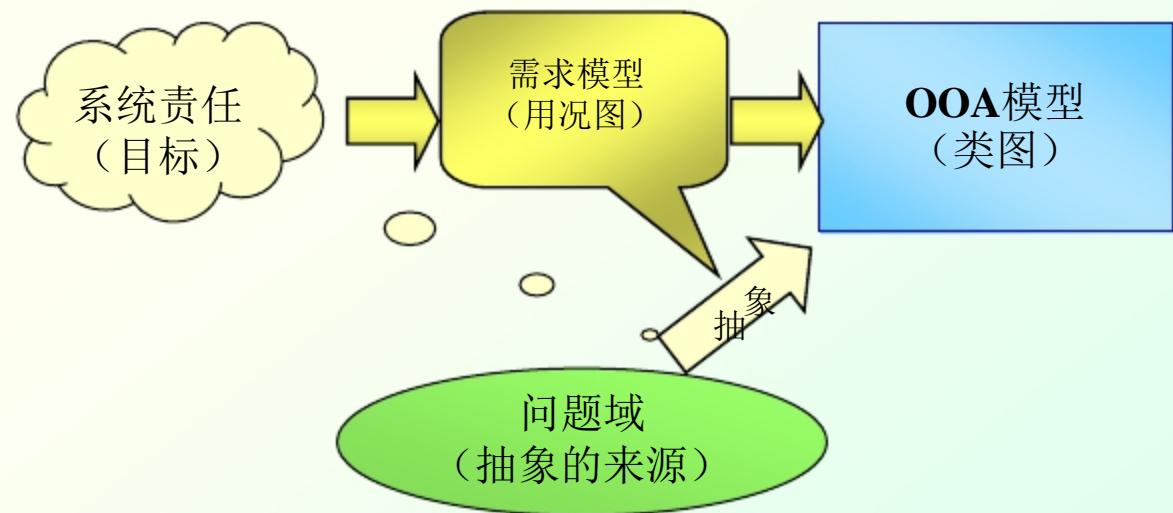
什么是参与者，有些种类

基于用况概念的需求模型对**OOA**的意义

用况的内容描述

用况图中的模型元素和表示法

如何建立用况图



第6~8章 建立类图

发现对象

定义属性与操作

定义各种关系

继承（一般-特殊结构）

聚合（整体-部分结构）

关联，消息

每一部分所用到的概念

表示法——类图中各种模型元素的画法

建模活动——策略与启发

审查与调整，异常情况处理

接口、依赖、关联端点修饰——简单了解

如何运用**OO**方法基本概念解决各种复杂问题

(在其他方法中用扩充概念解决的问题)

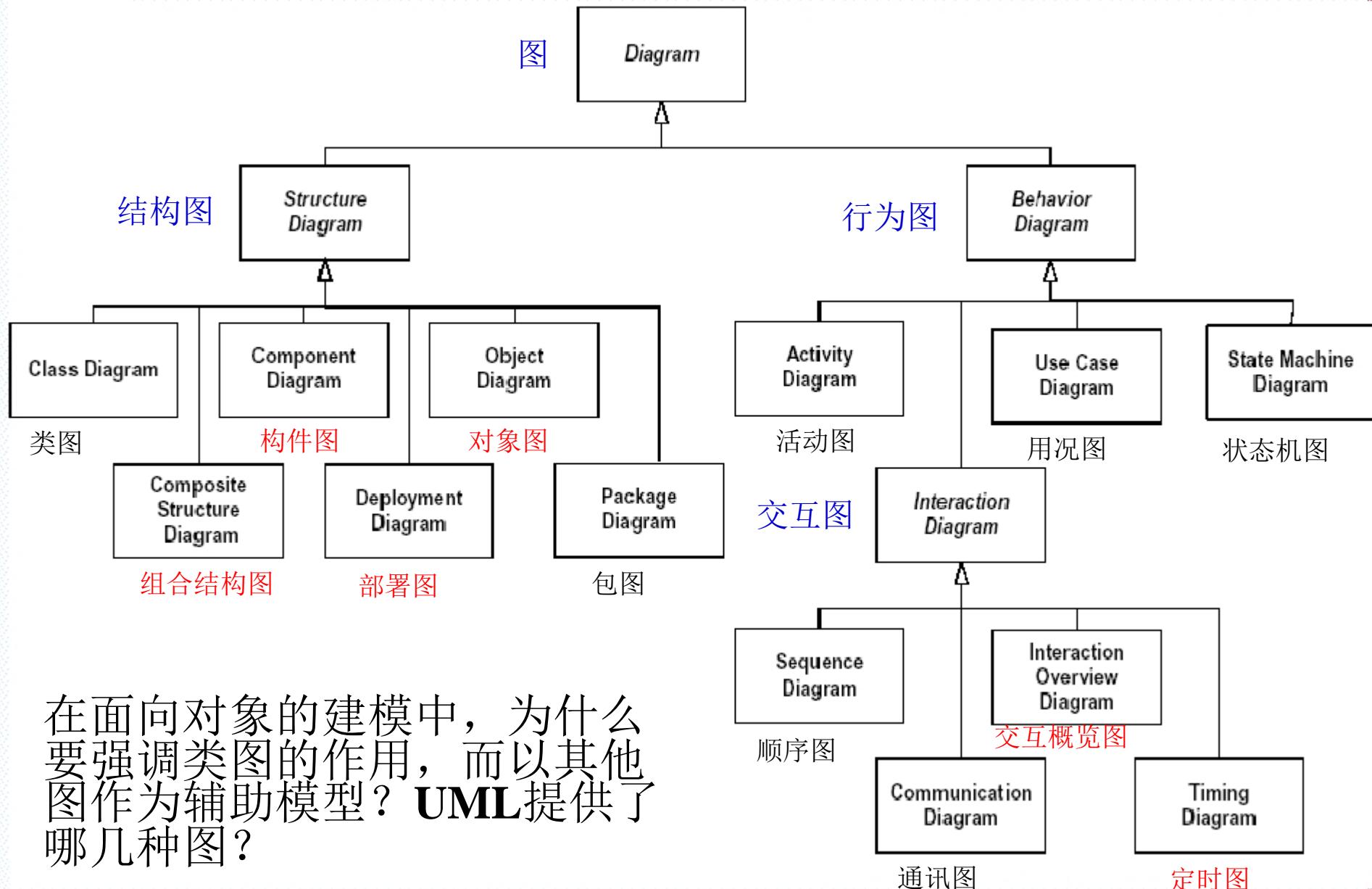
——关联类，多元关联，限定符，动态变化的对象特征

运用**OO**概念解决典型问题的技巧——书中的例子，例如：

利用一般-特殊结构支持软件复用

整体-部分结构的各种高级应用技巧

第9章 建立辅助模型



在面向对象的建模中，为什么要强调类图的作用，而以其他图作为辅助模型？UML提供了哪几种图？

重点掌握：

包图、顺序图、活动图、状态机图、通信图
以及基本模型——类图，需求模型——用况图
(其他各种图——简单了解)

每种图有什么用途

特别是对面向对象的建模有什么作用

每种图使用的主要概念，以及在图中的表示法

概念  表示法

第10章 建立模型规约

模型规约的作用

类规约包括哪些内容

如何组织模型规约使之有利于复用和机器处理

第11章 问题域部分的设计

什么是问题域部分？

它与OOA模型有什么关系？



将**OOA**模型搬到**OOD**
作为**OOD**的基础

按编程语言、网络
、操作系统、复用
支持等实现条件进
行必要的调整

第12章 人机交互部分的设计

什么是人机交互部分

把人机交互部分作为系统中一个独立的部分有什么好处

人机交互部分的需求分析

如何从用况提取人机交互

如何形成合理的命令组织结构

根据人机交互需求选择界面元素

运用**OO**概念和表示法表达所有的界面成分

第13章 控制驱动部分的设计

什么是控制驱动部分

相关技术问题

由系统总体方案决定的实现条件

软件体系结构——特别是分布式系统的软件体系结构

系统的并发性——进程与线程

如何设计控制驱动部分

选择软件体系结构风格

确定系统分布方案

对象的分布——类的分布——类图的分解

如何识别控制流

用主动类表示控制流

第14章 数据接口部分的设计

什么是数据接口部分

几种常用的数据管理系统

文件系统、**RDBMS**、**OODBMS**

用于对象存储时，各自的优点与缺点

用非**OO**的数据管理系统存储对象的指导思想