

问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响



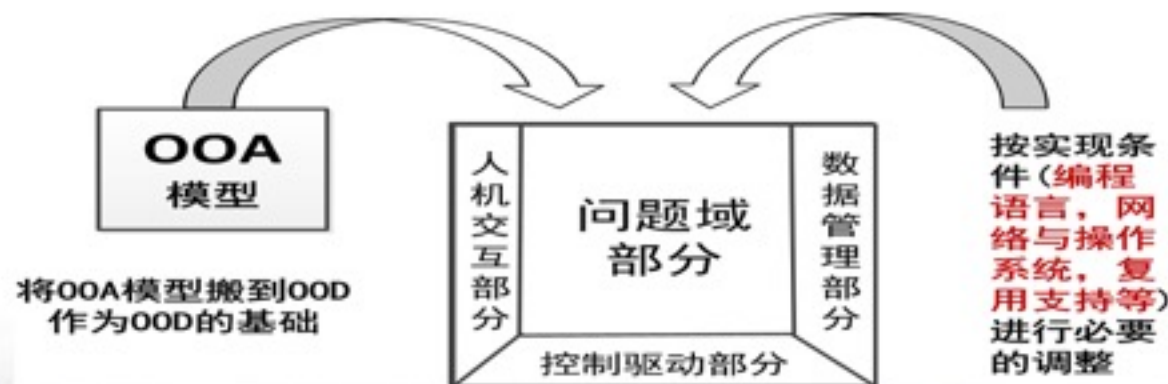
第三部分 问题域部分（PDC）的设计

一、什么是问题域部分？

问题域部分是OOD模型的四个组成部分之一，是由问题域有关的对象构成，并且在特定的实现条件下提供用户所需功能的组成部分。它是在OOA模型基础上按实现条件进行必要的修改、调整和细节补充而得到的。


不是传统方法的“转换”，不存在鸿沟。主要不是细化，但OOA未完成的细节定义要在OOD完成。

是对OOA模型的补充与调整。



问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响



并非所有的实现因素都能通过一些在OOD中新定义的独立组成部分而实行有效的隔离。有些实现因素将不可避免地影响到OOA阶段识别的对象，影响到它们的内部特征和相互关系，因而要求在OOD阶段按照这些条件对它们做必要的修改、调整和细节上的补充。这正是问题域部分的设计所要解决的问题。

- 1、编程语言
- 2、硬件、操作系统及网络设施
- 3、复用支持
- 4、数据管理系统
- 5、界面支持系统



问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响

二、为什么需要问题域部分的设计？

- ◆ 在OOA阶段只考虑问题域和系统责任，OOD则要考虑与具体实现有关的问题，需要对OOA结果的补充与调整；
- ◆ 使反映问题域本质的总体框架和组织结构长期稳定，而细节可变；
- ◆ 稳定部分（PDC）与可变部分（其它部分）分开，使系统从容地适应变化；
- ◆ 有利于同一个分析用于不同的设计与实现；
- ◆ 支持系统族和相似系统的分析设计及编程结果复用；
- ◆ 使一个成功的系统具有超出其生存期的可扩展性



问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响



三、实现条件对问题域部分的影响

现在分析和讨论各种实现条件将对OOD模型产生什么影响：

1、编程语言：

用于实现的编程语言对问题域部分的影响最大，其中包括两方面问题：

(1) 选定的编程语言可能不支持某些面向对象的概念与原则。此时要根据编程语言的实际表达能力对模型进行调整，以保证设计模型与源程序一致。

(2) OOA阶段可能将某些与编程语言有关的对象细节推迟到OOD阶段来定义。如对象的创建、删除、复制、转存、初始化等系统行为、属性的数据类型等。

编程语言确定之后，这些问题都要给出完整的解决。



问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响

2、硬件、操作系统及网络设施

选用的计算机、操作系统及网络设施对OOD的影响包括：对象在不同的站点上的分布、主动对象的设计、通信控制以及性能改进措施等。这些问题对问题域部分和控制驱动部分都将有影响。

3、复用支持

如果存在已经进行设计和编码的可复用类构件，用以代替OOA模型中新定义的类无疑将提高设计与编码效率。但这需要对模型做适当的调整与修改。

4、数据管理系统

选用的数据管理系统主要影响OOD模型的数据管理部分的设计，但也需要对问题域部分的某些类补充访问该数据管理部分所要求的属性与操作。



问题域部分的概念

- 什么是问题域部分？
- 为什么需要问题域部分的设计？
- 实现条件对问题域部分的影响

5、界面支持系统

指支持用户界面开发的软件系统。主要影响人机交互部分的设计，对问题域部分影响很少，只是两部分之间需要互传消息而已。

