

控制驱动部分的设计

- 什么是控制驱动部分？
- 为什么需要控制驱动部分？
- 如何设计控制驱动部分？



第五部分

控制驱动部分的设计

一、什么是控制驱动部分

控制驱动部分——

是OOD模型的组成部分之一，该部分由系统中全部主动类构成。这些主动类描述了整个系统中所有的主动对象，每个主动对象是系统中的一个控制流的驱动者。

控制流是一个在处理机上顺序执行的动作序列。在目前的实现技术中，一个控制流就是一个进程或线程。

在顺序程序中，只有一个控制流，并发程序则含有多个控制流。每个控制流开始执行的源头，是一个主动对象的主动操作。



控制驱动部分的设计

- 什么是控制驱动部分？
- 为什么需要控制驱动部分？
- 如何设计控制驱动部分？



二、为什么需要控制驱动部分

并发行为是现实中固有的

当前大量的系统都是并发系统（多任务系统），例如：

- 设备与计算机并发工作的系统
- 有多个窗口进行人机交互的系统
- 多用户系统
- 多个子系统并发工作的系统
- 单处理机上的多任务系统
- 多处理机系统

.....

多任务的设置

- 描述问题域固有的并发行为
- 表达实现所需的设计决策

为了隔离硬件、操作系统、网络的变化对整个系统的影响



北京大学



控制驱动部分的设计

- 什么是控制驱动部分？
- 为什么需要控制驱动部分？
- 如何设计控制驱动部分？

三、如何设计控制驱动部分

1、识别控制流

1) 以节点为单位识别控制流

不同节点上的程序之间的并发，以及同一节点上的程序间的并发。

2) 从用户需求出发识别控制流

3) 从use case认识控制流

OOA阶段定义的每一个use case都描述了一项独立的系统功能。从需求角度，它描述了一项系统功能的业务处理流程；从系统构造来看，它很可能暗示需要通过一个控制流来实现其业务处理流程。

4) 参照OO模型中的主动对象

主动对象的一个主动操作是一个控制流的源头。



控制驱动部分的设计

- 什么是控制驱动部分？
- 为什么需要控制驱动部分？
- 如何设计控制驱动部分？

5) 为改善性能而增设的控制流

高优先级任务：把对时间要求较高的工作从其他工作中分离出来，作为独立的任务，用专门设计的控制流去实现。

低优先级任务：可把这些工作分离出来，作为独立的任务，用专门设计的控制流去实现，在执行时赋予较低的优先级，使它们称为通常所说的后台进程。

紧急任务：把这些工作作为单独的任务，用专门的控制流去实现。它的执行不允许其他任何任务干扰。

6) 实现并行计算的控制流

7) 实现节点间通信的控制流

某些情况下，为了实现的方便，可能要设计一些专门负责与其他节点通信的控制流。

8) 对其他控制流进行协调的控制流

- 可以设计一个主进程，由它负责系统的启动和初始化，其他进程的创建和撤销、资源分配、优先级的授予等工作；
- 也可以把负责协调的控制流设计成一个控制流，而把其他控制流设计成它内部的线程。





控制驱动部分的设计

- 什么是控制驱动部分？
- 为什么需要控制驱动部分？
- 如何设计控制驱动部分？

2、审查与筛选

- 去掉不必要的控制流
 - 多余的并发性意味着执行效率的损失
 - 每个控制流应该有以上列举的理由之一
 - 不要人为地增加控制流

3、控制流的表示与分类

每个控制流都可以用主动对象的一个主动操作来描述

• 在主动对象的表示中区别进程和线程：

UML中，在类的名字栏中内附加一个标准衍型《process》或《thread》，以表明这个类描述的是进程还是线程。

• 主动操作的表示：

UML2.0未提供主动操作的表示法，如果想表示主动操作，可以在操作前加《active》



北京大学

数据管理部分的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



第六部分

数据管理部分的设计

一、什么是数据管理部分

数据管理部分是OOD模型中负责与具体的数据管理系统衔接的外围组成部分，它为系统中需要长久存储的对象提供了在选定的数据管理系统中进行数据存储与恢复的功能。

不同的数据管理系统：

文件系统、R-DBMS、OO-DBMS

——对数据管理部分的设计有不同的影响

问题范围：

对象在永久性存储介质上的存储

只存储对象的属性部分

可能只有一部分对象需要长久存储

二、为什么需要数据管理部分

为了隔离数据管理系统对其它部分的影响

使选用不同的数据管理系统时，问题域部分基本相同



清华大学

数据管理部分的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



第六部分

数据管理部分的设计

一、什么是数据管理部分

数据管理部分是OOD模型中负责与具体的数据管理系统衔接的外围组成部分，它为系统中需要长久存储的对象提供了在选定的数据管理系统中进行数据存储与恢复的功能。

不同的数据管理系统：

文件系统、R-DBMS、OO-DBMS

——对数据管理部分的设计有不同的影响

问题范围：

对象在永久性存储介质上的存储

只存储对象的属性部分

可能只有一部分对象需要长久存储

二、为什么需要数据管理部分

为了隔离数据管理系统对其它部分的影响

使选用不同的数据管理系统时，问题域部分基本相同



清华大学

数据管理部分的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



三、如何设计数据管理部分

1、选择数据管理系统

文件系统，R-DBMS，OO-DBMS

2、数据存放设计

针对文件系统：

(1) 对象存放策略

用文件系统存放对象的基本策略是：把由每个类直接定义，并需要永久存储的全部对象实例，放在一个文件中；其中每个对象实例的全部属性作为一个存储单元，占用该文件的一个记录。

具有继承关系的类的存放策略，让通过一般类直接创建的对象实例和通过特殊类创建的对象实例分别使用不同的文件，以保持文件中每个记录是等长的，并且每个记录中都没有空余不用的字节。



北京大学

数据管理部分的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



(2) 设计数据管理部分的对象类：

一个最主要的对象类是为所有（需要在文件中存储数据的）其它对象提供基本保存与恢复功能的对象类，可将它命名为“对象存储器”。应用系统中各个类的对象是按关键字存取，还是按对象名称存取，还是两者兼而有之，这将对“对象存取器”类的设计提出不同的要求。

(3) 问题域部分的修改


问题域部分的对象通过请求数据管理部分提供的操作实现对象的保存 恢复。为了实现这种请求，这些对象类需要增加一些属性和操作。

对每个需要长久保存其对象实例的对象类，要增加一个属性“类名”，使它的对象在发出请求时以该属性的值作为参数，指出自己是属于哪个类的；数据管理部分通过它知道应该对哪个文件进行操作。



数据管理部分 的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



此外，要增加一个“请求保存”操作和一个“请求恢复”操作，它们的功能是向数据管理部分的“对象存储器”对象发消息，分别请求后者的“对象保存”操作和“对象恢复”操作，从而把自己当前的状态（属性值）保存到文件中，或者从文件中恢复以往保存的结果。

由于每个需要长久保存其对象实例的类都需要上述属性和操作，因此可以增加一个一般类来定义它们，供所有这样的类继承。



数据管理部分 的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？

针对R-DBMS：

(1) 对象在数据库中的存放策略：

用关系数据库存放对象的策略是：把由每个类直接定义并需要永久存储的全部对象实例存放在一个数据库表中。

——每个类使用一个数据库表

- 1) 列出每个类的所有属性（包括继承来的属性）
 - 2) 规范化——一般按第三范式，
 - 3) 定义数据库的表按时间与空间权衡
- 列：规范化之后的一个属性
行：一个对象实例



数据管理部分的设计

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？



(2) 设计数据管理部分的类 并修改问题域部分 ——两种方案

方案1：

问题域部分：每个类的对象自己存储自己

数据管理部分：设立一个对象，提供两个操作——

- (1) 通知问题域部分的对象自己存储自己
- (2) 检索被存储的对象

为了存储自己，对象要知道什么？

本类对象的属性数据结构

本类对象对应哪个数据库表

对象实例对应数据库表的哪一行



北京大学

数据管理部分的设计



方案2：

数据管理部分设立一个对象，负责问题域部分所有对象的存储与检索

问题域部分的对象通过消息使用数据管理部分对象的操作

- 什么是数据管理部分？
- 为什么需要数据管理部分？
- 如何设计数据管理部分？

为了存储各个类的对象，数据管理部分的对象要知道什么？

每个要求存储、检索的类的属性数据结构

每个要求存储、检索的类的对象存放在哪个数据库表

当前要求存储或检索的对象属于哪个类，对应数据库表的哪一行

