



天津大学  
Tianjin University

## 第2章 软件体系结构建模





### ❁ 软件体系结构建模的种类

- ◎ 结构模型
- ◎ 框架模型
- ◎ 动态模型
- ◎ 过程模型
- ◎ 功能模型



### ❁ 软件体系结构建模的种类 – 结构模型

- 最直观、最普遍的建模方法。这种方法以体系结构的构件、连接件和其他概念来刻画结构，并力图通过结构来反映系统的重要语义内容，包括系统的配置、约束、隐含的假设条件、风格、性质等。
- 研究结构模型的核心是体系结构描述语言。



### ❁ 软件体系结构建模的种类 – 框架模型

- 框架模型与结构模型类似，但它不太侧重描述结构的细节而更侧重于**整体的结构**。
- 框架模型主要以一些特殊的问题为目标建立只针对和适应该问题的结构。



### ❁ 软件体系结构建模的种类 – 动态模型

动态模型是**对结构或框架模型的补充**，研究系统的“大颗粒”的行为性质。例如，描述系统的重新配置或演化。动态可以指系统总体结构的配置、建立或拆除通信通道或计算的过程。





### ❁ 软件体系结构建模的种类 – 过程模型

- 过程模型研究构造系统的**步骤和过程**。
- 结构是遵循某些过程脚本的结果。



### ❁ 软件体系结构建模的种类 – 功能模型

- 功能模型认为体系结构是由一组功能构件按层次组成，下层向上层提供服务。
- 功能模型可以看作是一种特殊的框架模型。



### ❁ 4+1模型 – 概述

- Kruchten在1995年提出了4+1的视图模型。
- 4+1视图模型从5个不同的视角包括**逻辑视图**、**进程视图**、**物理视图**、**开发视图**和**场景视图**来描述软件体系结构。
- 每一个视图只关心系统的一个侧面，5个视图结合在一起才能反映系统的软件体系结构的全部内容。





### ❁ 4+1模型 – 概述

最终用户：功能需求

逻辑视图

编程人员：软件管理

开发视图

场景

进程视图

系统集成人员：性能  
可扩充性、吞吐量等

物理视图

系统工程人员：系统  
拓扑、安装、通信等

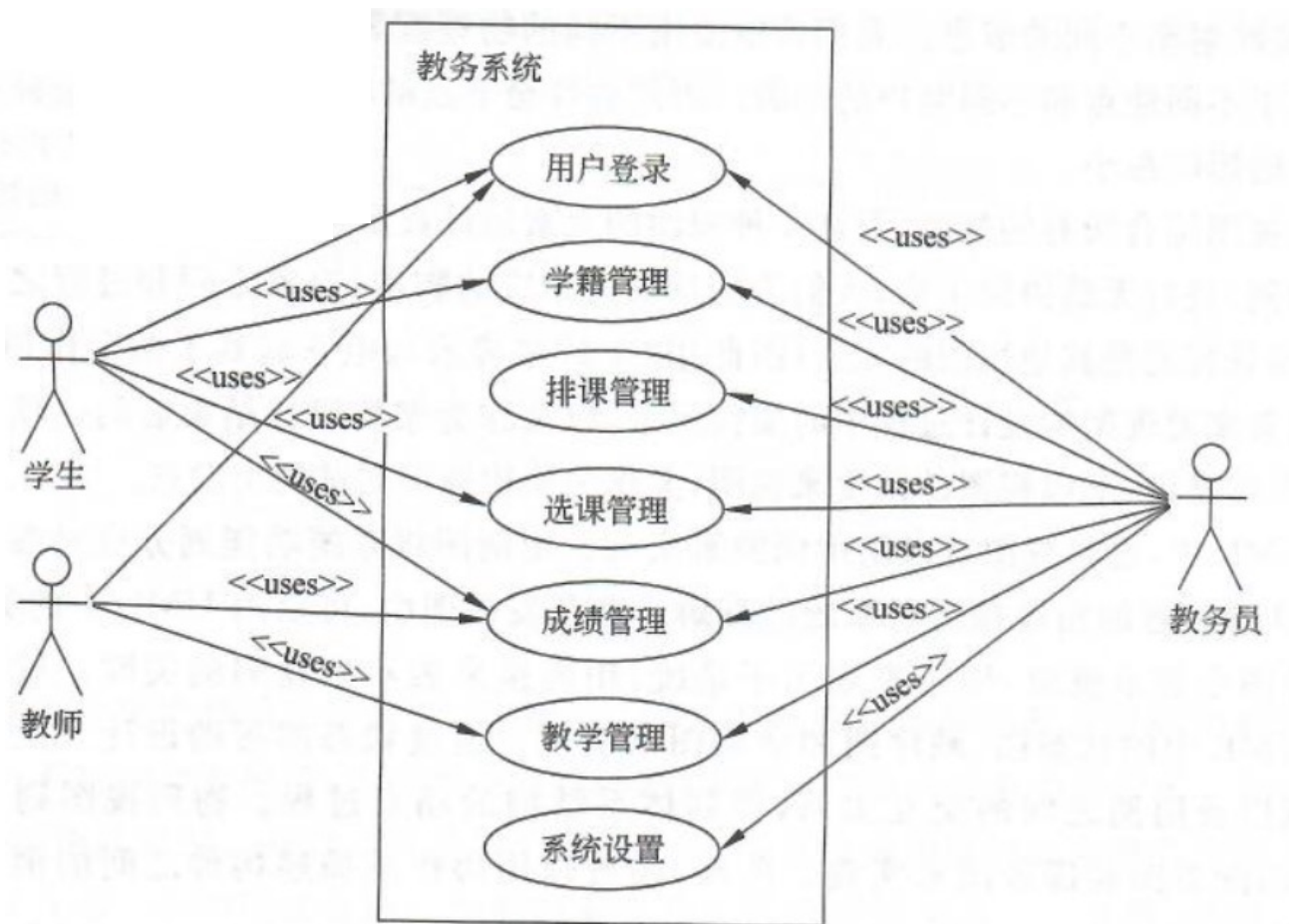


### ❁ 4+1模型 – 逻辑视图

- 逻辑视图主要支持系统的**功能需求**，即系统提供给最终用户的服务。  
在逻辑视图中，系统分解成一系列的功能抽象，这些抽象主要来自问题领域。这种分解不但可以用来进行功能分析，而且可用作标识在整个系统的各个不同部分的通用机制和设计元素。
- 在面向对象技术中，通过抽象、封装和继承，可以用对象模型来代表逻辑视图，用类图来描述逻辑视图。

## 第2章 软件体系结构建模

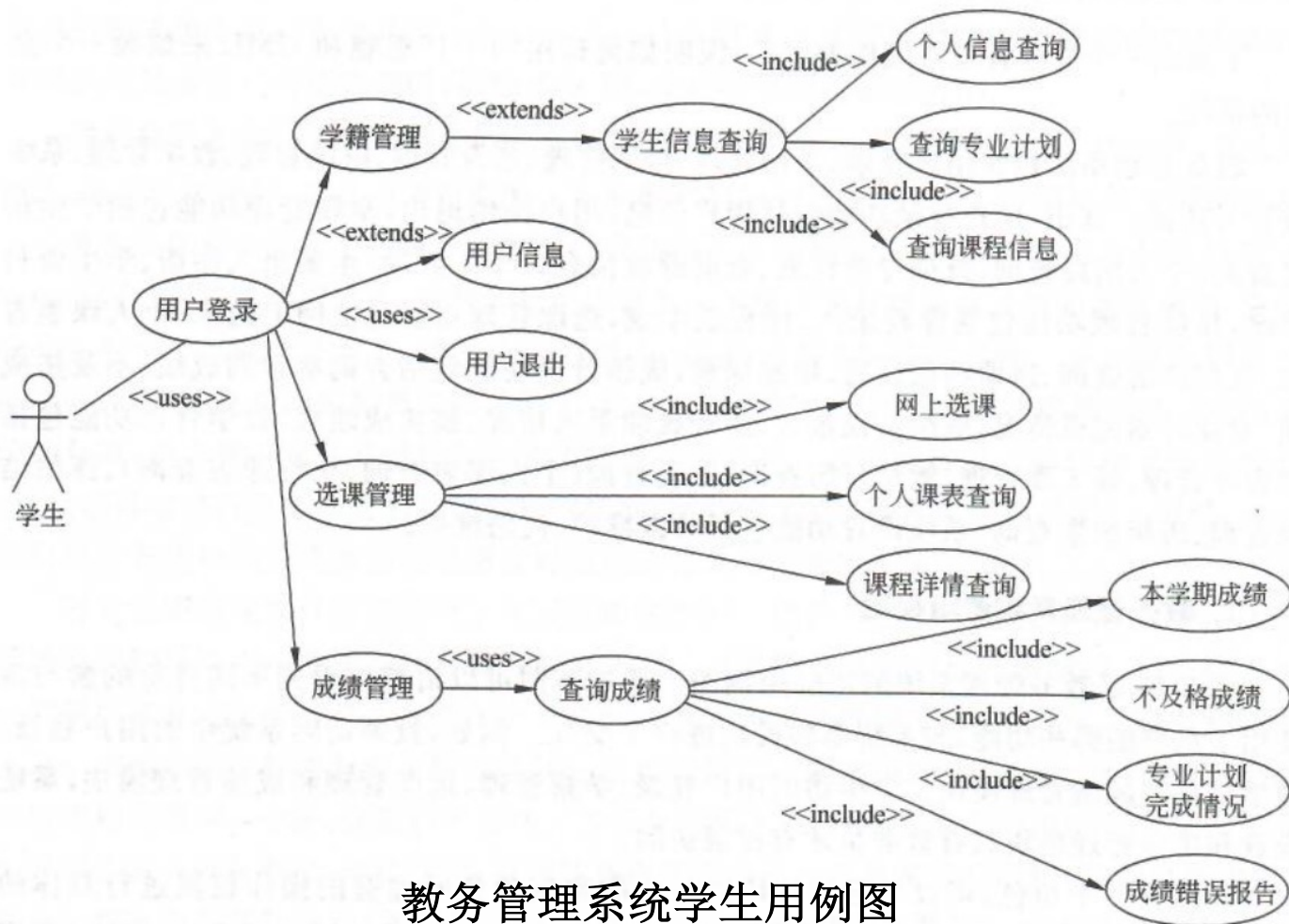
### ❁ 4+1模型 – 逻辑视图



教务管理系统顶层用例

## 第2章 软件体系结构建模

### 4+1模型 - 逻辑视图



教务管理系统学生用例图



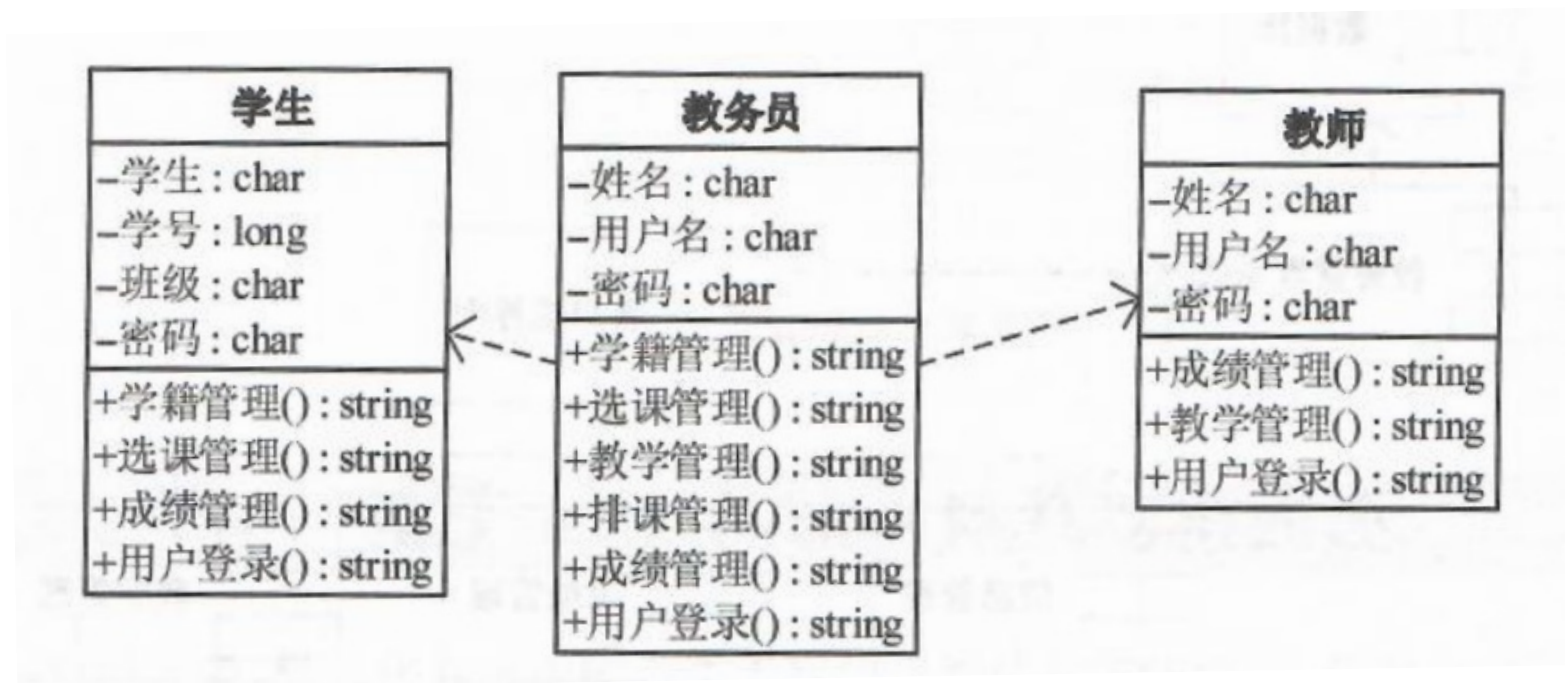
### ❁ 4+1模型 – 开发视图

- 开发视图也称模块视图，主要侧重于软件模块的组织和管理。
- 开发视图要考虑软件内部的需求，如软件开发的容易性、软件的重用和软件的通用性，要充分考虑由于具体开发工具的不同而带来的局限性。
- 开发视图通过系统输入输出关系的模型图和子系统图来描述。



## 第2章 软件体系结构建模

### ❁ 4+1模型 – 开发视图

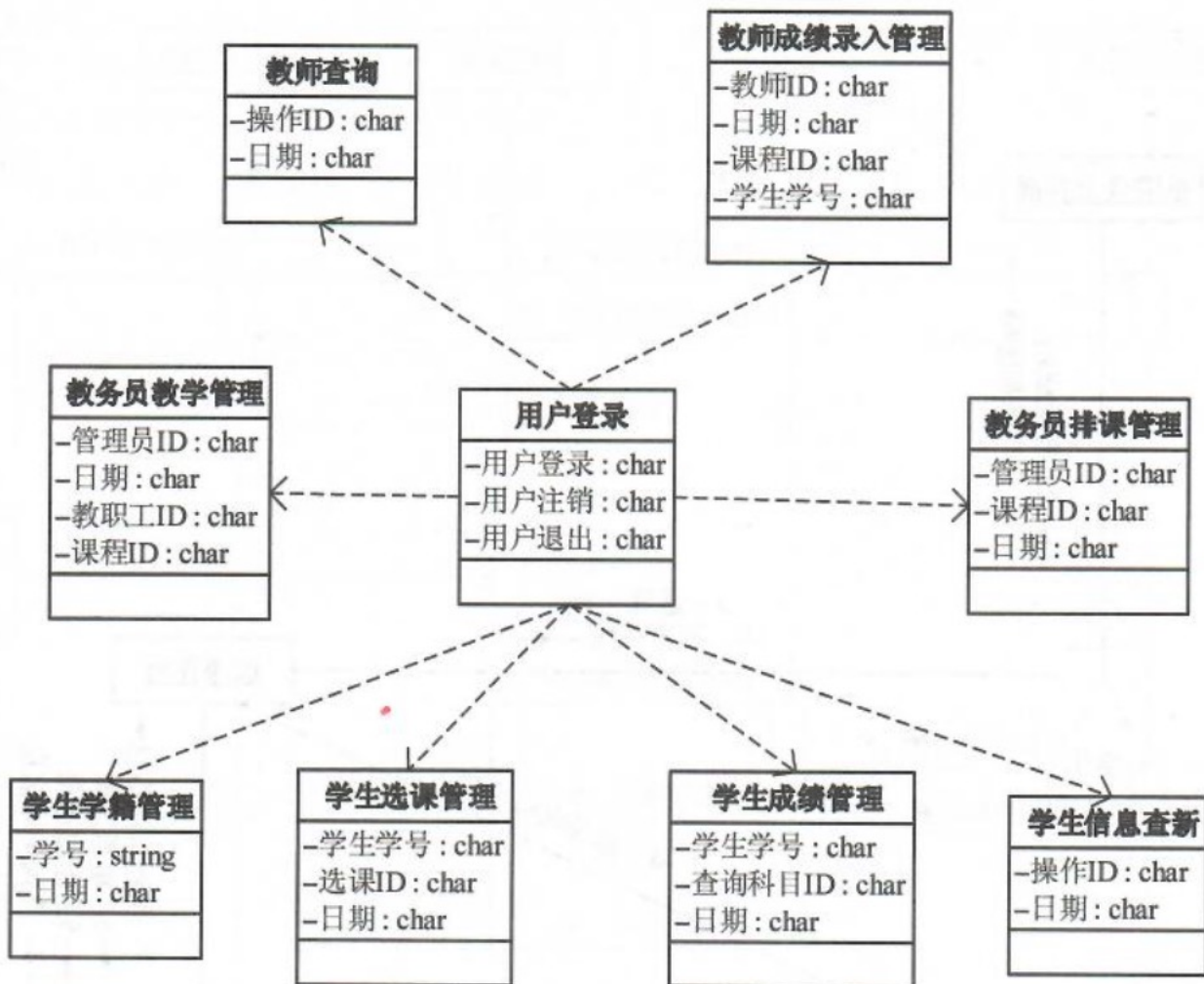


教务管理系统人员类图



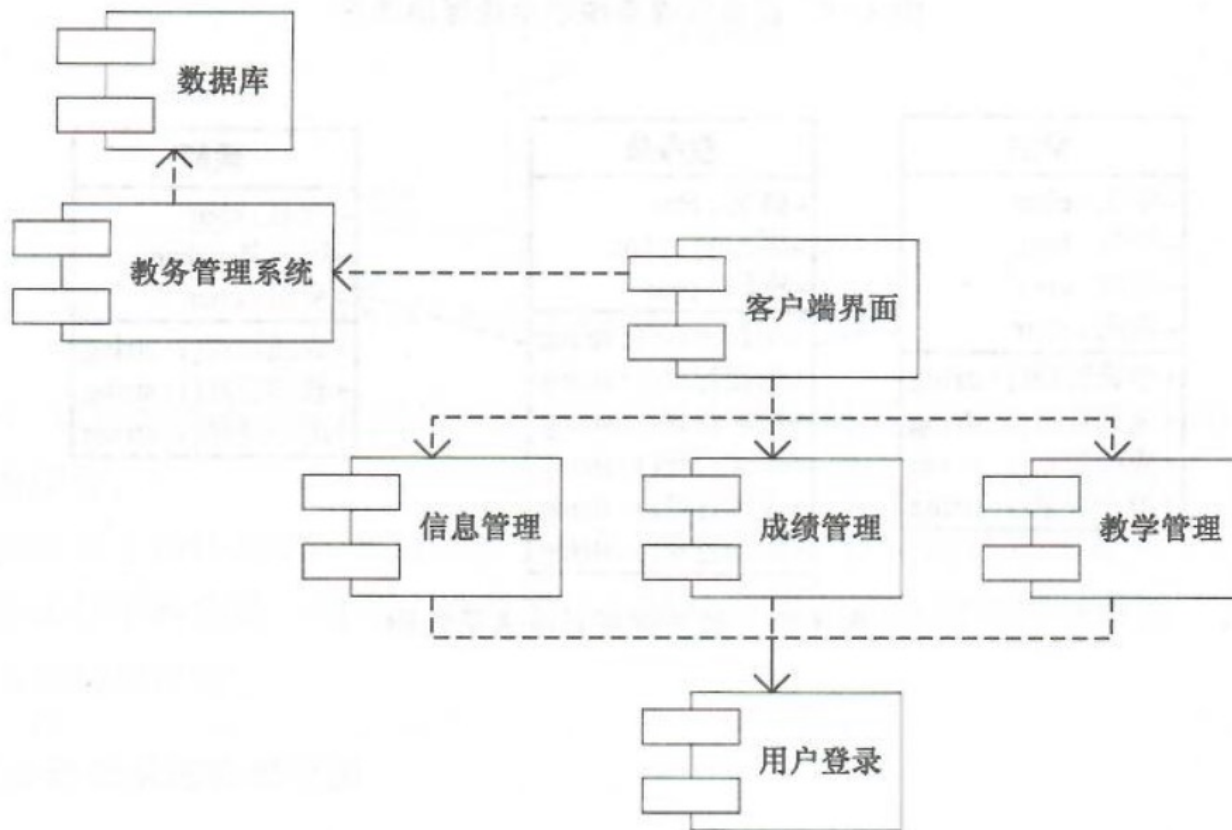
## 第2章 软件体系结构建模

### 4+1模型 - 开发视图



## 第2章 软件体系结构建模

### ❁ 4+1模型 – 开发视图



教务管理系统构件图



### ❁ 4+1模型 – 开发视图

- 在开发视图中，最好采用4-6层子系统，而且每个子系统仅仅能与同层或更低层的子系统通讯，这样可以使每个层次的接口既完备又精练，避免了各个模块之间很复杂的依赖关系。
- 设计时要充分考虑，对于各个层次，层次越低，通用性越强，这样，可以保证应用程序的需求发生改变时，所做的改动最小。开发视图所用的风格通常是层次结构风格。

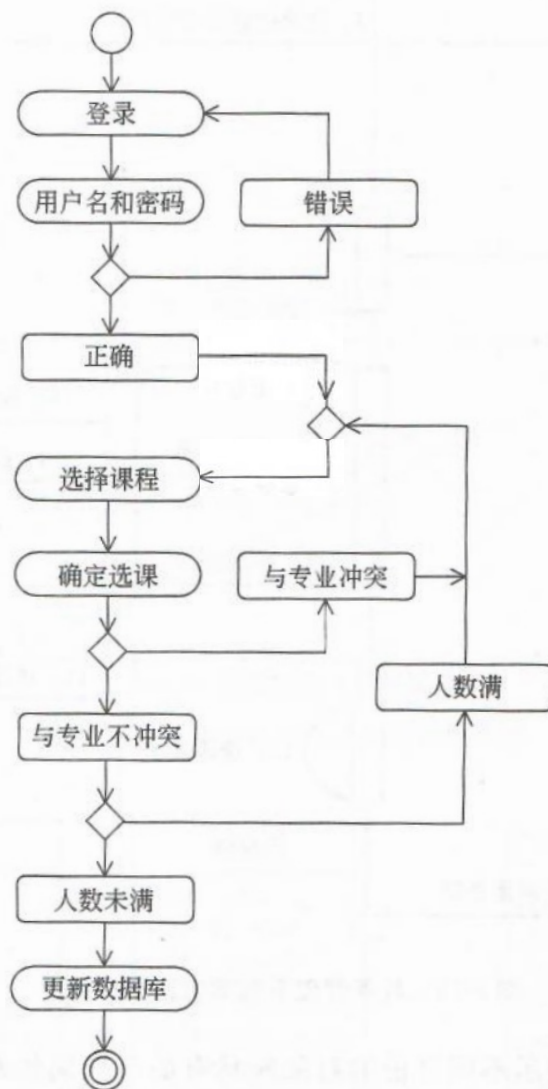


### ❁ 4+1模型 – 进程视图

- 进程视图侧重于系统的**运行特性**，主要关注一些非功能性的需求。
- 进程视图强调**并发性、分布性、系统集成性和容错能力**，以及从逻辑视图中的主要抽象如何适合进程结构。它也定义逻辑视图中的各个类的操作具体是在哪一个线程中被执行的。
- 进程视图可以描述成多层抽象，每个级别分别关注不同的方面。在最高层抽象中，进程结构可以看作是构成一个执行单元的一组任务。它可看成一系列独立的，通过逻辑网络相互通信的程序。它们是分布的，通过总线或局域网、广域网等硬件资源连接起来。

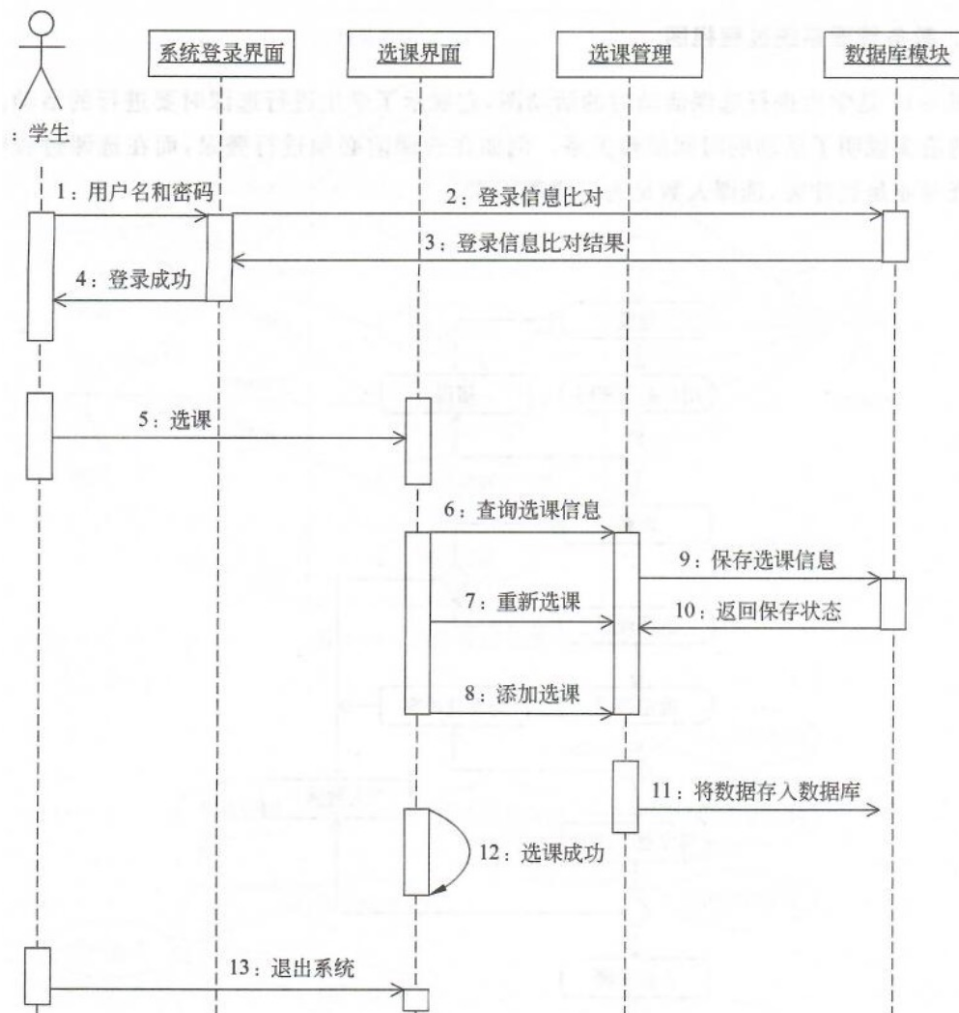
## 第2章 软件体系结构建模

### ❁ 4+1模型 - 进程视图



## 第2章 软件体系结构建模

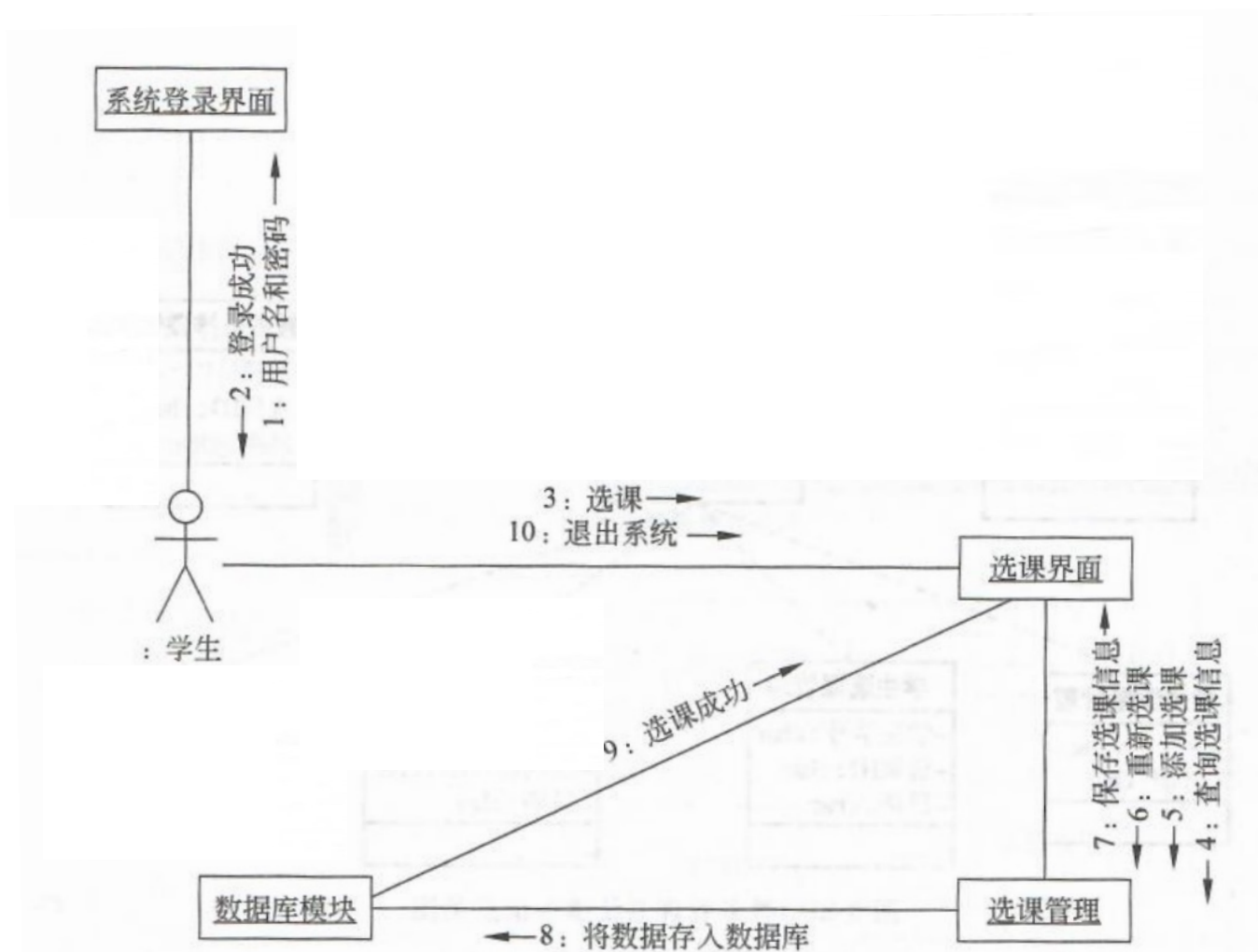
### ❁ 4+1模型 – 进程视图





## 第2章 软件体系结构建模

### ❁ 4+1模型 – 进程视图



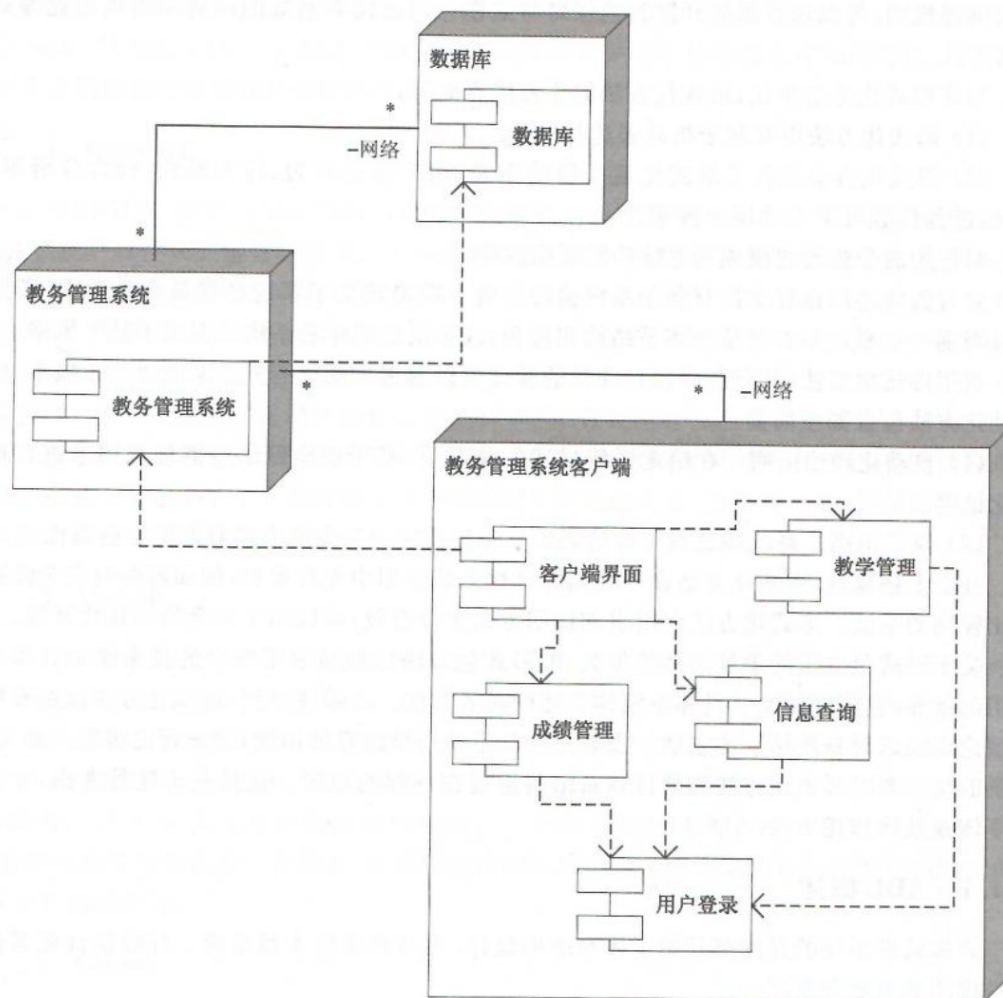


### ❁ 4+1模型 – 物理视图

- 物理视图主要考虑如何把软件映射到硬件上，它通常要考虑到系统性能、规模、可靠性等。解决系统拓扑结构、系统安装、通讯等问题。
- 当软件运行于不同的节点上时，各视图中的构件都直接或间接地对应于系统的不同节点上。因此，从软件到节点的映射要有较高的灵活性，当环境改变时，对系统其他视图的影响最小。

## 第2章 软件体系结构建模

### ❁ 4+1模型 – 物理视图





### ❁ 4+1模型 – 场景

- 场景可以看作是**那些重要系统活动的抽象**，它使四个视图有机联系起来，从某种意义上说场景是最重要的需求抽象。在开发体系结构时，它可以帮助设计者找到体系结构的构件和它们之间的作用关系。同时，也可以用场景来分析一个特定的视图，或描述不同视图构件间是如何相互作用的。
- 场景可以用文本表示，也可以用图形表示。

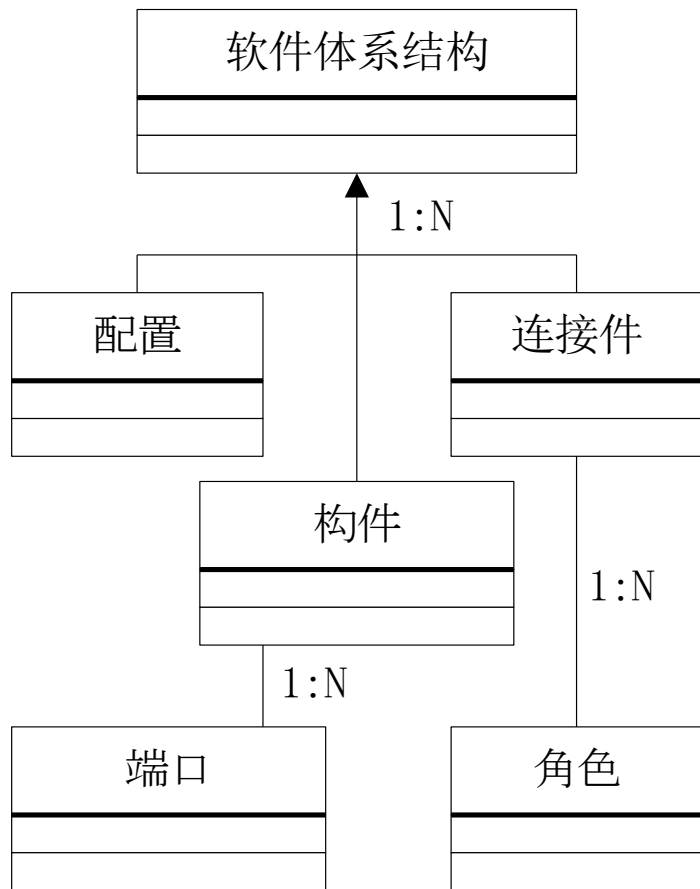


### ❁ 4+1模型 – 小结

- 逻辑视图和开发视图描述系统的静态结构，而进程视图和物理视图描述系统的动态结构。
- 对于不同的软件系统来说，侧重的角度也有所不同。例如，对于管理信息系统来说，比较侧重于从逻辑视图和开发视图来描述系统，而对于实时控制系统来说，则比较注重于从进程视图和物理视图来描述系统。



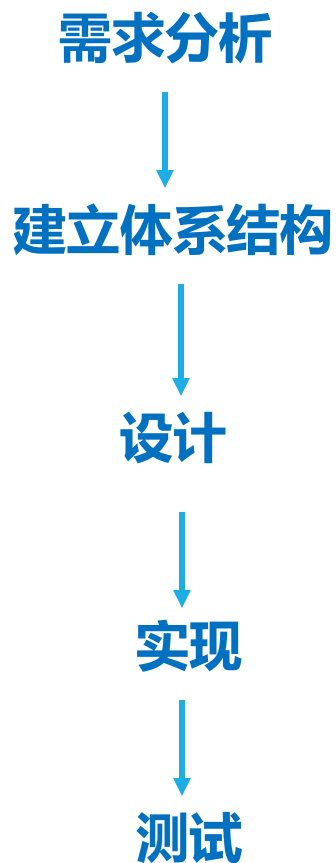
### ❁ 软件体系结构的核心模型







### ❁ 软件体系结构的生命周期模型



### 软件体系结构的生命周期模型

