

# 第03讲 统一建模语言

**徐峰磊**

**QQ:474017512**

Copyright©2022, Software Research Team in USTS

All rights reserved

# 主要内容

1

建模语言类别  
(Type)

2

UML的特点  
(Characteristic)

3

案例图示  
(Case)



# 3.1 建模语言三个类别

- 系统建模提供了我们用计算机思考现实系统的一种方法，如何表示成我们计算机软件开发人员能够看懂并使用的模型，建立的模型需要用正确的语言来表达，在语言的表达形式上，分成三类：
  - 非形式化的
  - 半形式化的
  - 形式化的

UML是一种半形式化的建模语言,是系统建模的标准。

# 模型与建模

- 软件系统是对现实需求的计算机程序模拟，建模是为软件开发建立蓝图描绘各阶段的计划、以及收集整理各阶段的制品（输出）。
- 软件模型是对计算机软件系统的所有表达，它是对软件系统的一切描述，包括软件过程中的各阶段的各种描述文档、规约、图形表示体系、编码阶段的所有代码、交付给用户的最终产品以及技术手册等。
- 在解决问题之前必须首先理解所要解决的问题。对问题理解得越透彻，就越容易解决它。为了更好地理解问题，人们常常采用建立问题模型的方法。
- 所谓模型，就是为了理解事物而对事物作出的一种抽象

# 为什么要建模

- 建立大厦和建立茅草屋的区别在于前者有计划和目的性
- 模型可以帮助人们对已有遗产系统的理解
- 模型的建立加深对新系统 in 开发过程中不断深入的理解
- 模型可以重用（文档重用、经验重用）

# 误区一：建模就等于是写文档

■建模很象是作计划：作计划的价值在于计划编制的过程中而非计划本身，而计划体现为一种文档。

■“模型”与“文档”这二者在概念上是有关系的，可以认为，模型体现为文档，但模型的价值体现在建立模型的过程之中而非模型本身。

■模型始终处于修改增加删除的动态变化过程中，最后得到的模型不能体现其中的迭代修改过程。因此，建模的过程才是最有价值的。

■总之、价值体现在建模的活动中，而非模型本身。

## 误区二：从开始阶段解决一切问题

- 在一开始不可能考虑和预见到所有的细枝末节。
  - 不论在实现阶段之前的各子阶段的规约多完备，但实现阶段的代码会很快地与之失去同步。
  - 可运行的计算机系统最终是由代码实现的，实现的技术细节决定了最初的规约说明需要妥协技术可行性的调整。

# 讨论

- 存在需求规约和代码的背离吗？
- 为什么最终交付的产品往往需要返工，这说明什么？
- 如何解决误区二提出的问题？



# 解决误区二提出的问题

- 需求规约描述系统是什么，没有涉及实现的任何技术细节。从这个定义上来说它和代码毫无关系，**不存在**是否与代码背离的问题。
- 最终交付的产品不能达到需求规约的要求，确实又是代码这个软件系统最终的实体和需求规约**产生了**背离。
- 需求规约未能为系统实现准确达标；
- 用户需求未能精确表达为需求规约。虽然需求规约是具有法律效应的，但依法办事何其难哉！用户是上帝与需求规约的合同相比，前者似乎更占上风。
- 迭代/增量的软件过程模型，some、more、even more ...持续频繁的交付集成可以降低失败风险，增大用户反馈。

# 误区三：所有的开发人员都知道如何建模

- 面临一个严重的问题：

- 许多不是开发人员的人，包括高级经理和用户，不知道软件企业的从业人员应该如何分工
- 不能够区分开高级程序员和一般程序员，认为所有的开发人员都应具备从头到尾开发整个系统的技能。

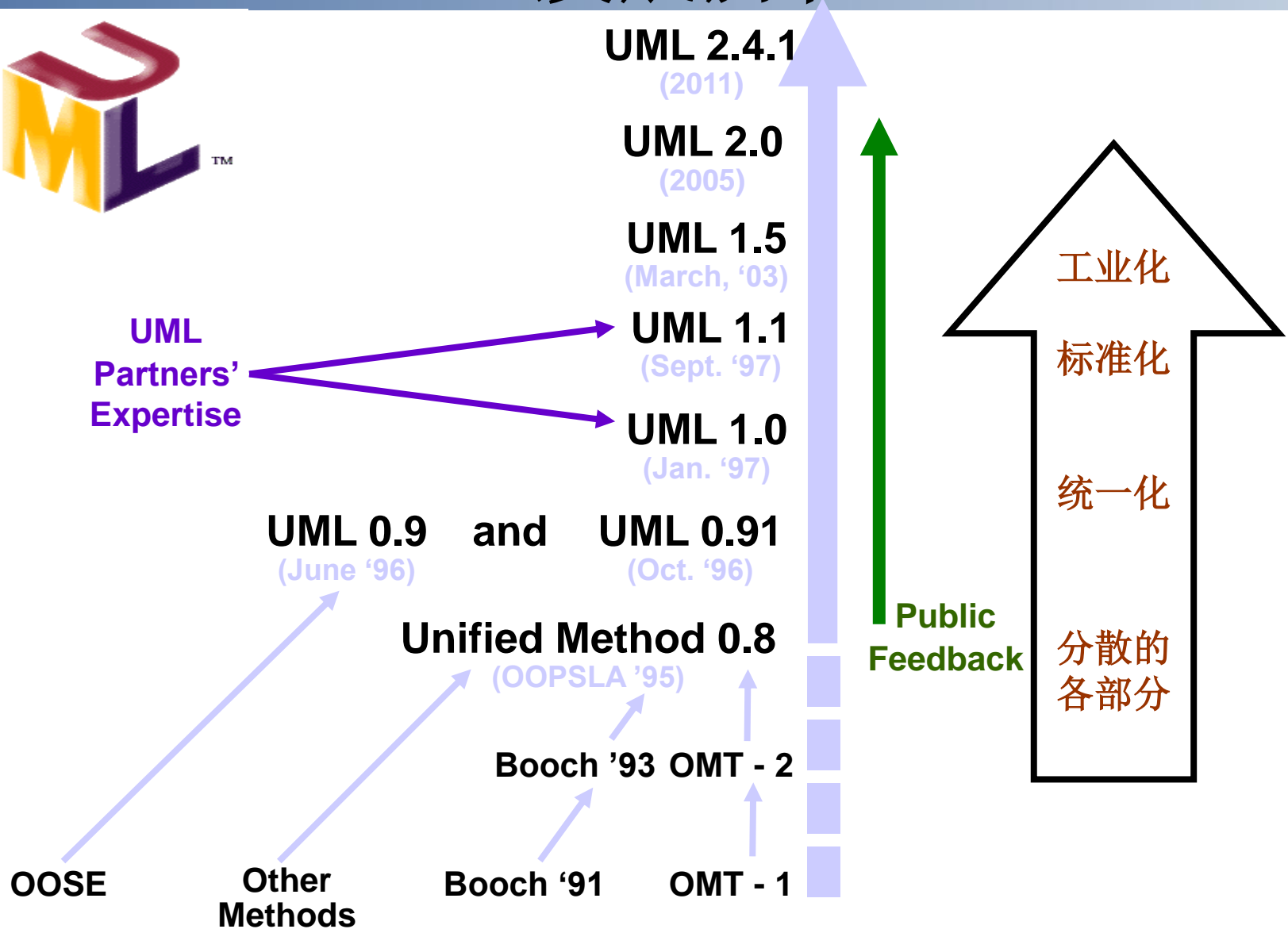
# 软件建模的原则

- (1) 准确的原则
- (2) 标准规范的原则
- (3) 子系统划分的原则（横向划分）
- (4) 分层原则（纵向划分）

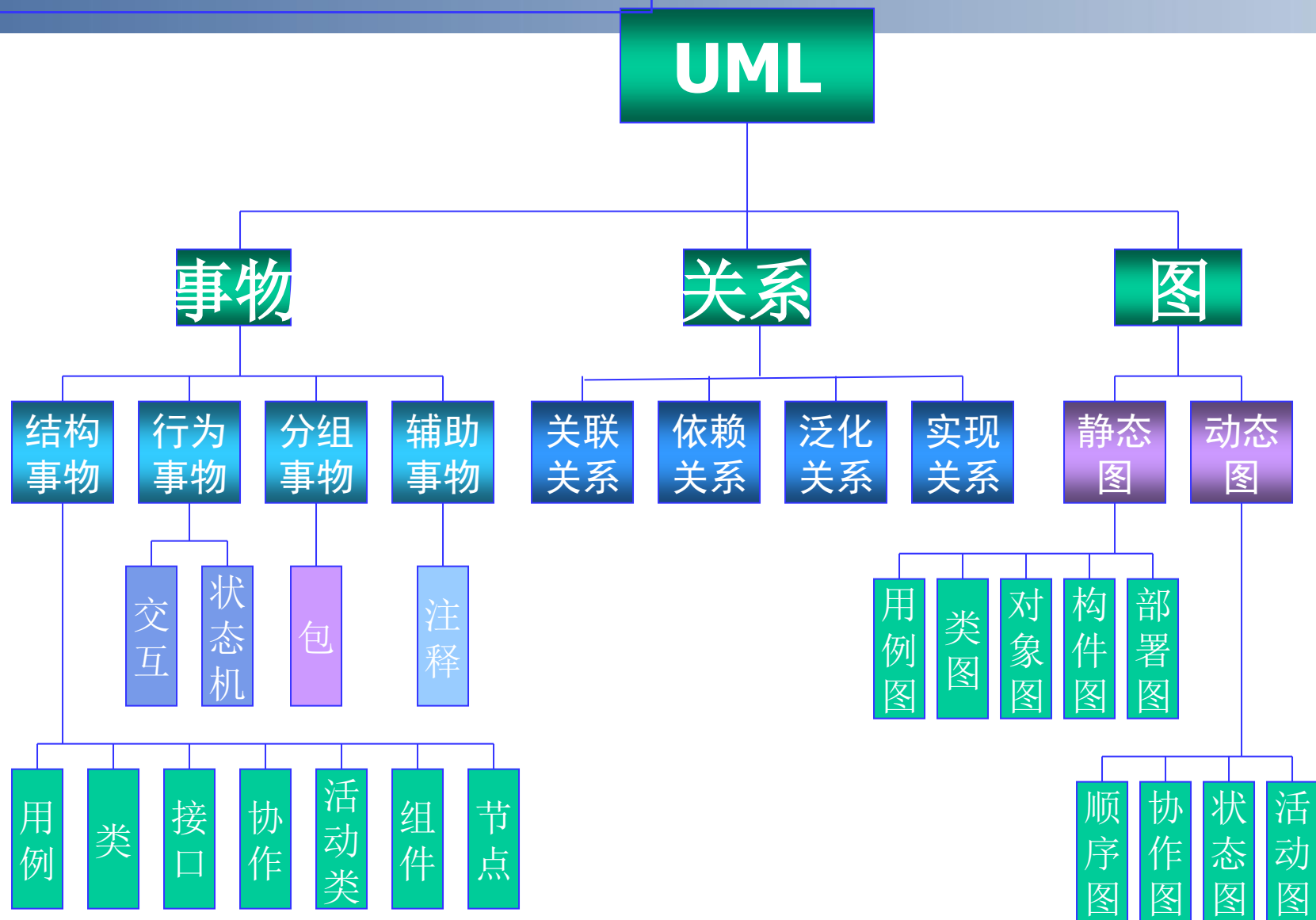
## 3.2 UML特点

- UML语言的主要特点有以下3点：
- (1) 统一了Booch、OMT和OOSE等基本概念。
- (2) UML还吸取了面向对象技术的优点，当然也有非面向对象的影响。
- (3) UML还提出了许多新说法。增加了模板、职责、扩展机制、线程、过程、分布式、并发、模式、合作、活动等概念，并清晰地地区分类型、类和实例、细化、接口和组件等概念。

# UML发展历程



# UML的组成



模型名称	与传统软件工程对应的模型	UML 图	说 明
用例模型	功能模型（使用工具：数据流图）	用例图	从用户角度描述系统需求，是所有开发活动的指南，即产生系统功能
静态模型	数据模型（使用工具：E-R图）	类图，对象图，构件图，部署图	描述系统的元素与元素间的关系
动态模型	行为模型（使用工具：状态转换图）	状态图，顺序图，协作图，活动图	描述系统随时间发展的行为

## 3.3 网络教学系统案例UML简单图示

- 案例主要展示UML九种图的表示方式，小型案例不一定需要九种图，有时有用例图和类图就足以能说明软件的主要功能和软件结构了。该案例主要展示UML的九种图样式，更多的UML细节我们在接下来的学习中详细解释说明。

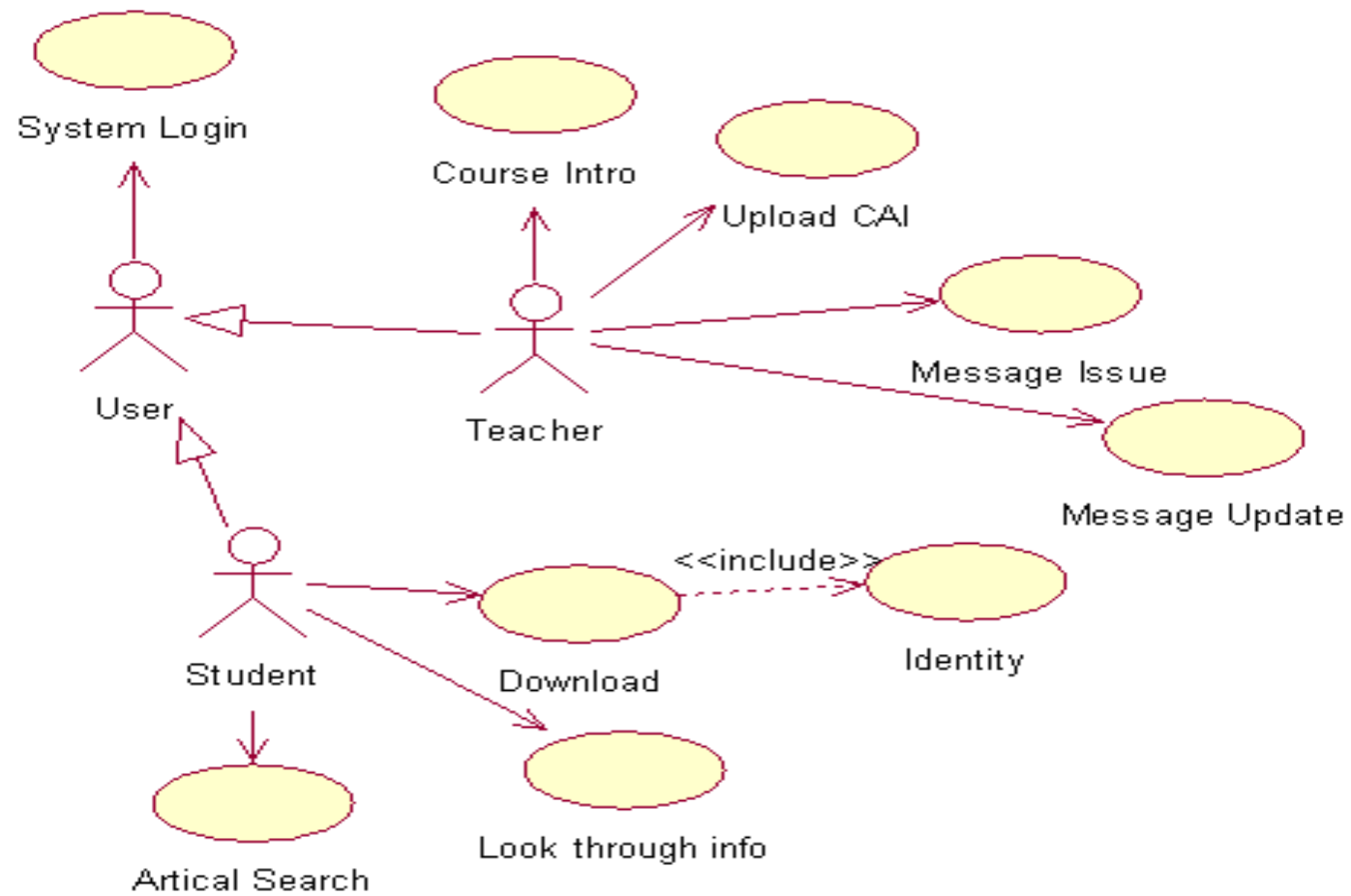


## 3.3.1 系统功能

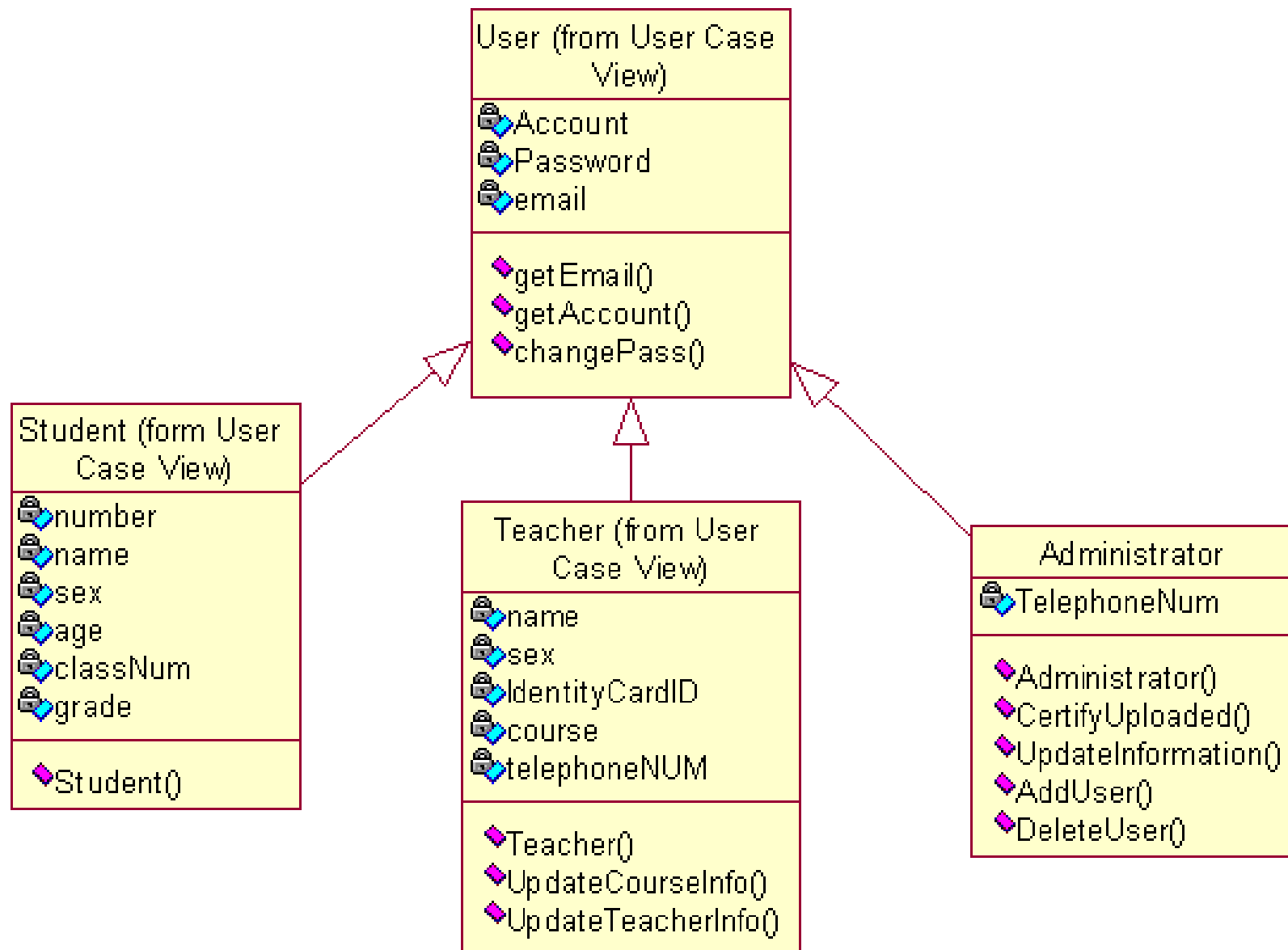
- 1、系统功能需求
  - (1) 学生可以登陆网站浏览和查找各种信息以及下载文件。
  - (2) 教师可以登陆网站给出课程见解、发布、修改和更新消息以及上传课件。
  - (3) 系统管理员可以对页面进行维护和批准用户的注册申请。

## 3.3.2 系统的UML建模

- 1、系统的用例图
- 创建用例图之前首先需要确定参与者。
- 在网络教学系统中，需要学生和教师的参与。学生可以浏览课程简介，教学计划，学习方法等教师发布的文章，并可以根据关键字查询文章。此外，学生可以从网站上下载课件。教师作为教学的主导者，使用此网站可以发布学习方法，课程的重点等和教学相关的文章，以及和课程相关的通知等，还可以将某一门课程的课件上传。
- 网站需要一个专门的管理者进行日常维护与管理，所以需要有一个系统管理员的参与。

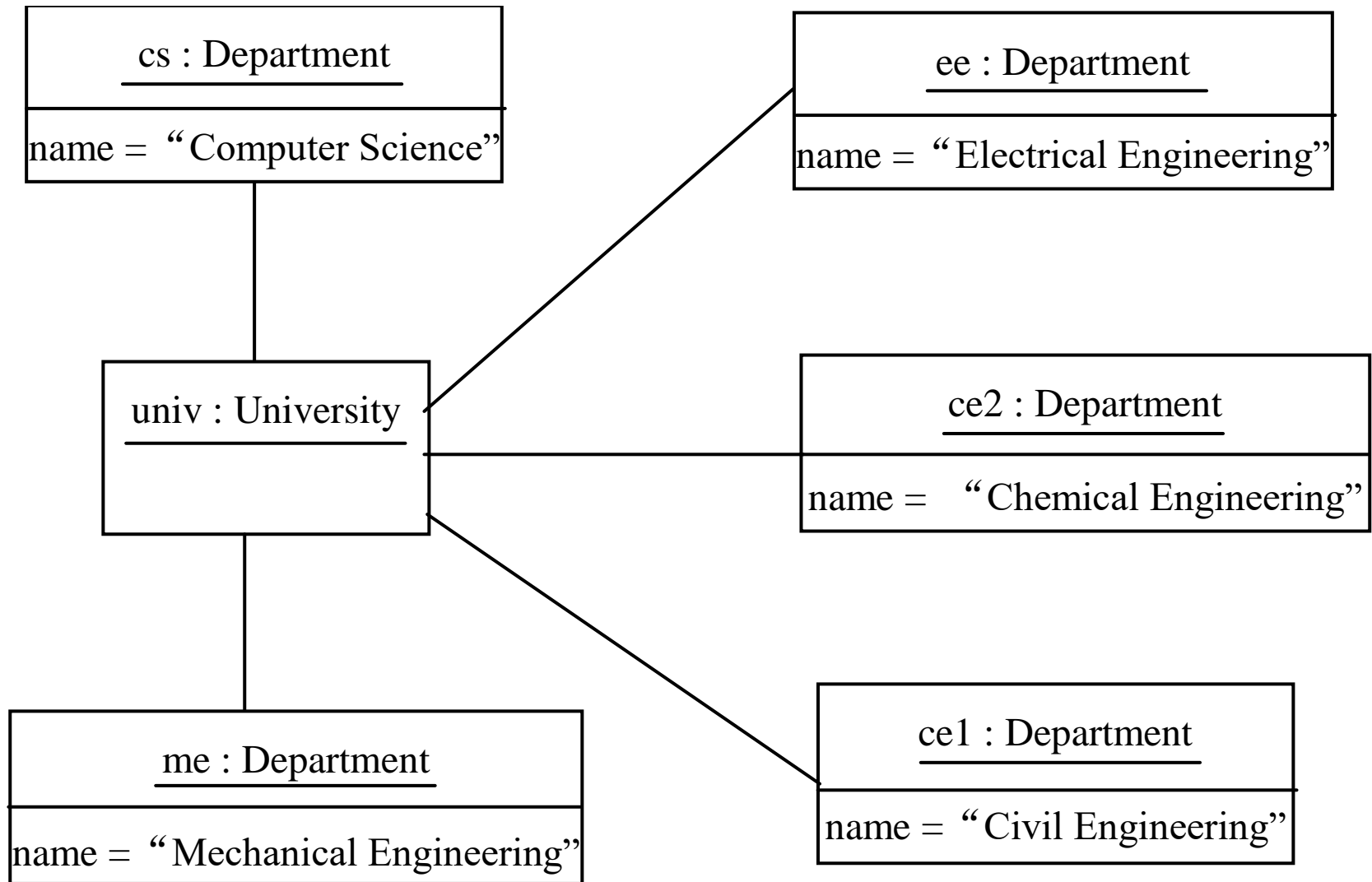


- 6、系统中的类
- （1）参与者相关的类
- 系统中和参与者相关的类的类图如下

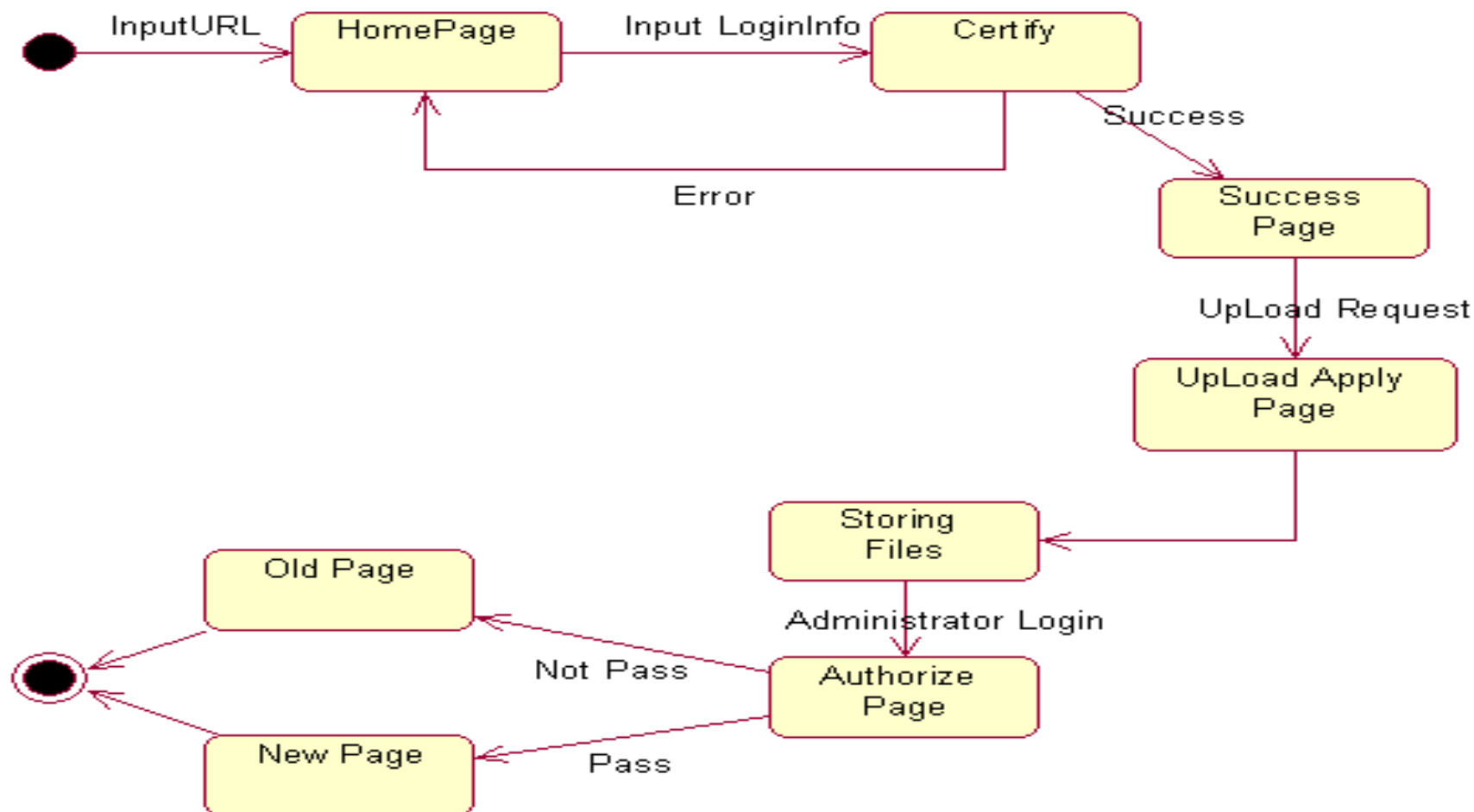


# 对象图

- 对象图（Object Diagrams）描述了某一瞬间对象集及对象间的关系。为处在时域空间某一点的系统建模，描绘了系统的对象、对象的状态及对象间的关系。
- 对象图主要用来为对象结构建模。对象图中通常含有：对象、连接。像其他的图一样，对象图中还可以有注解、约束、包或子系统。

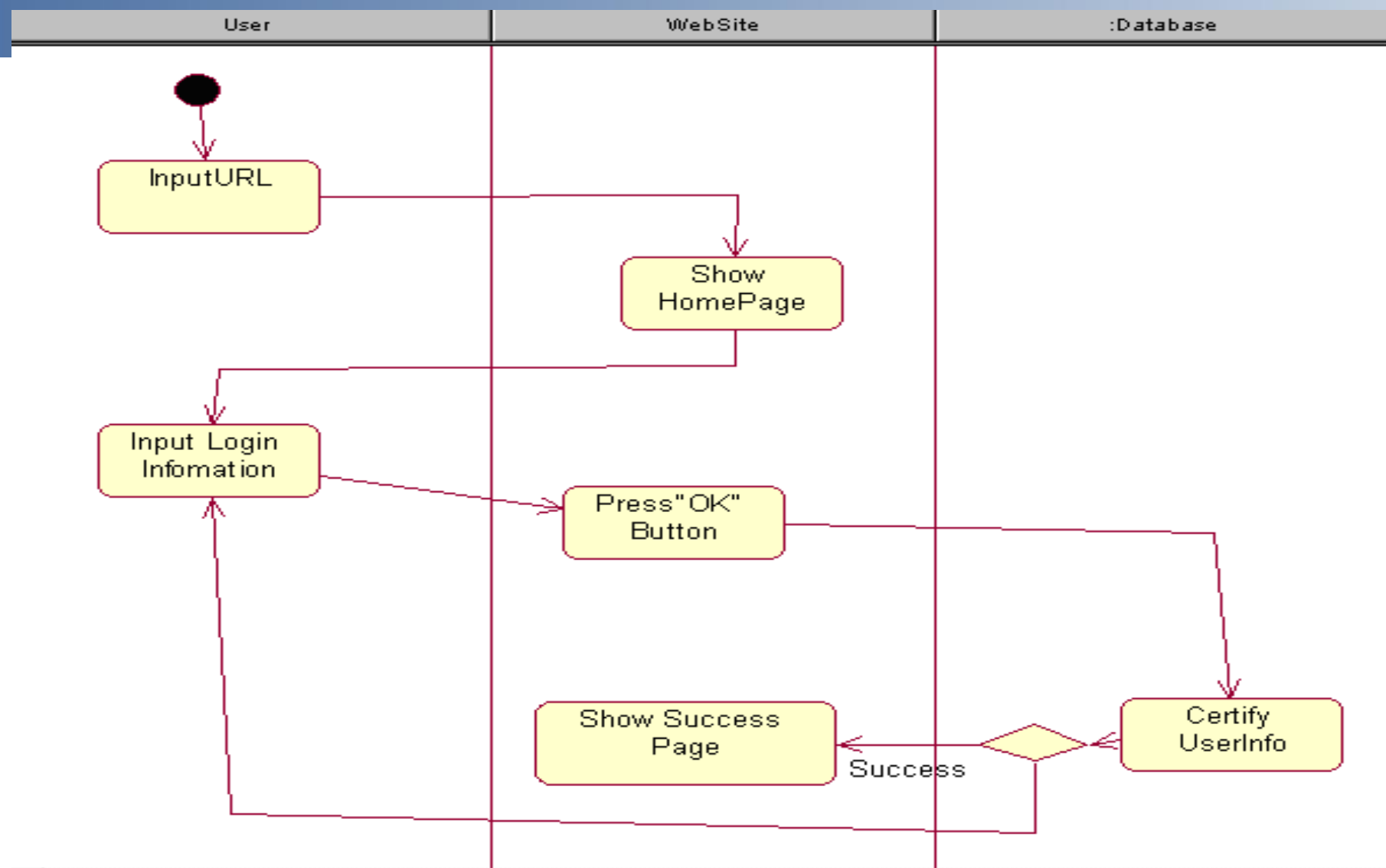


## ●4、系统的状态图：





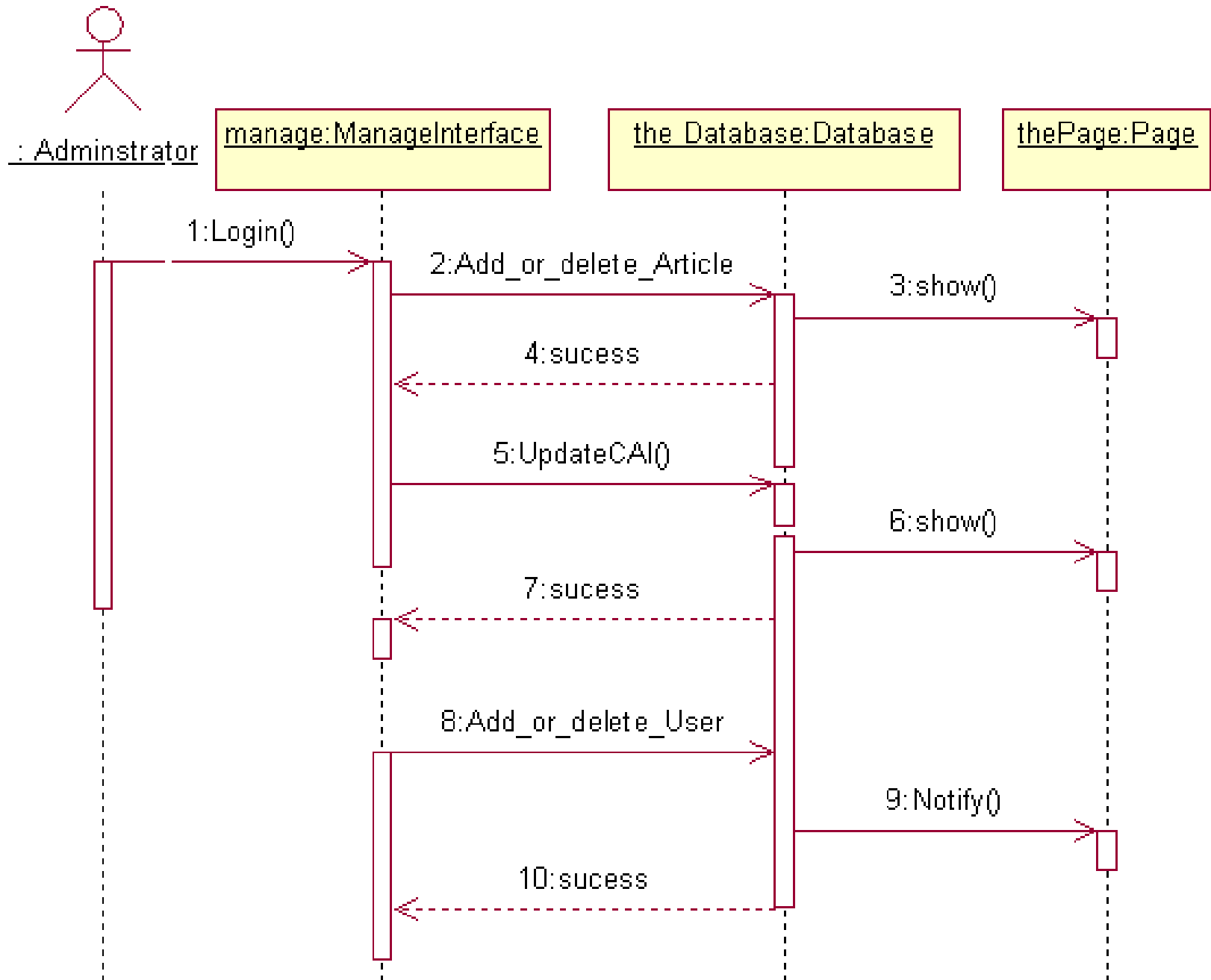
- 5、系统的活动图：
- （1）用户登录系统的活动图



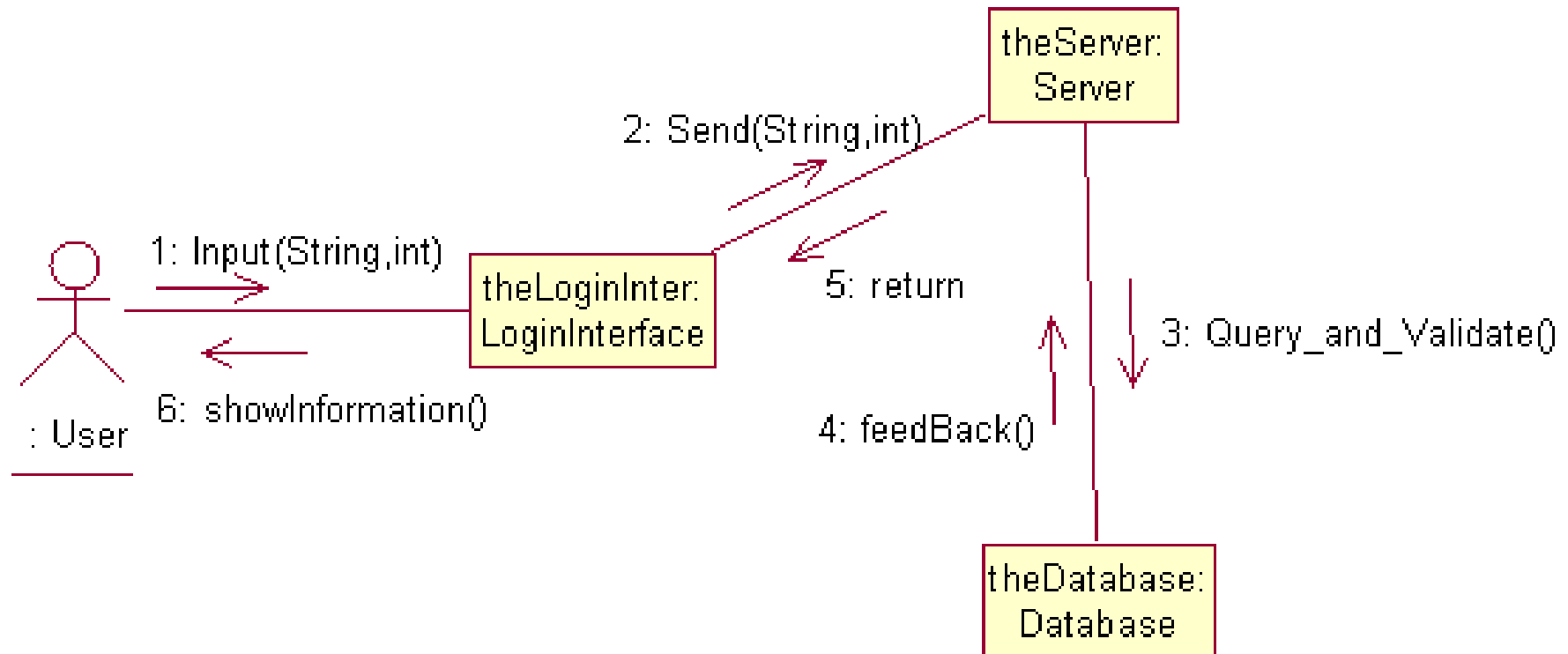
- 6、系统的时序图

- 网络教学系统中的用例很多，所能画出的时序图也很多，在此不一一介绍。

- （1）系统管理人员管理网站的时序图

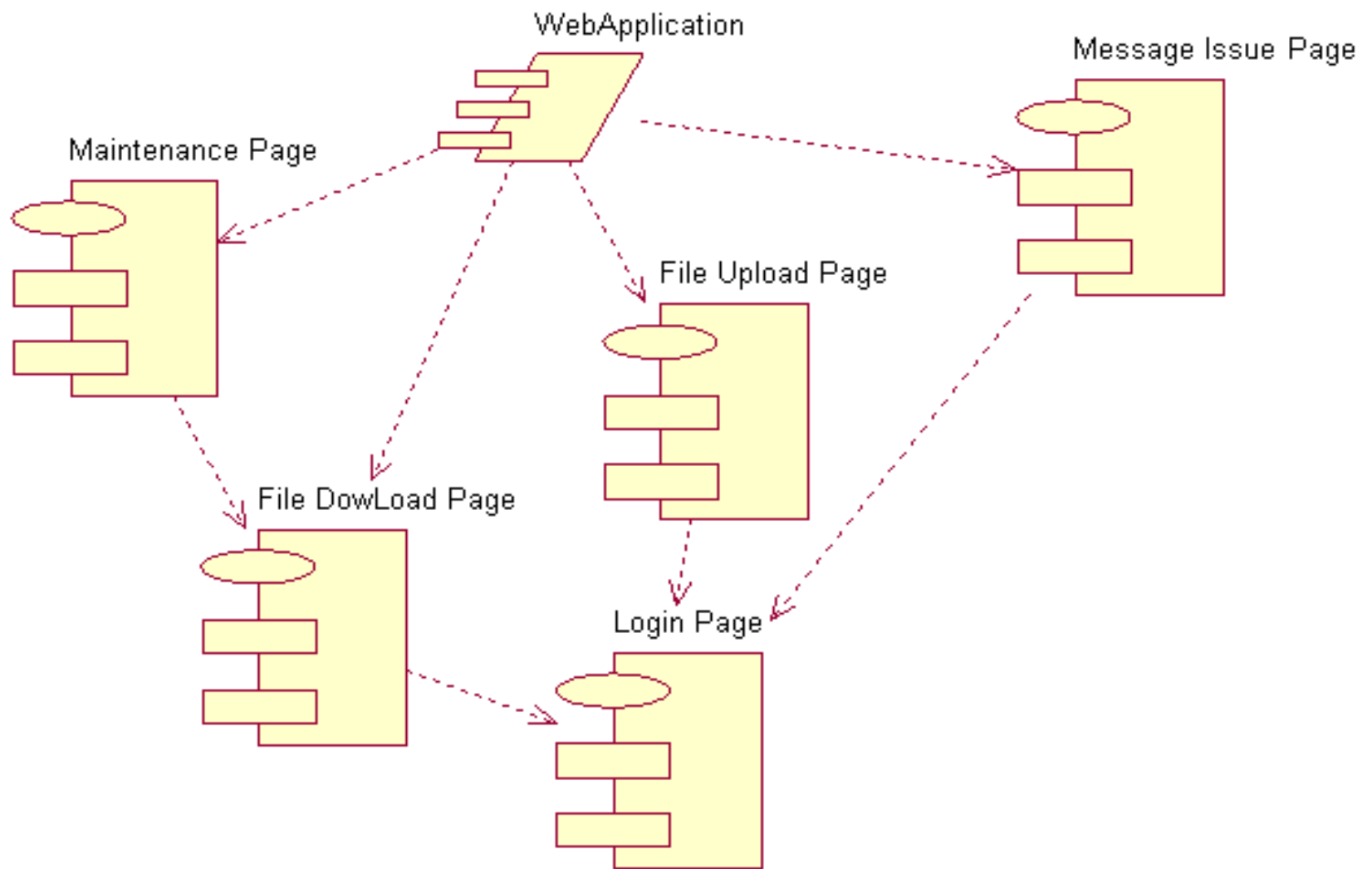


- 7、系统的协作图：
- 1、用户登录系统的协作图



## ●8、系统的组件图

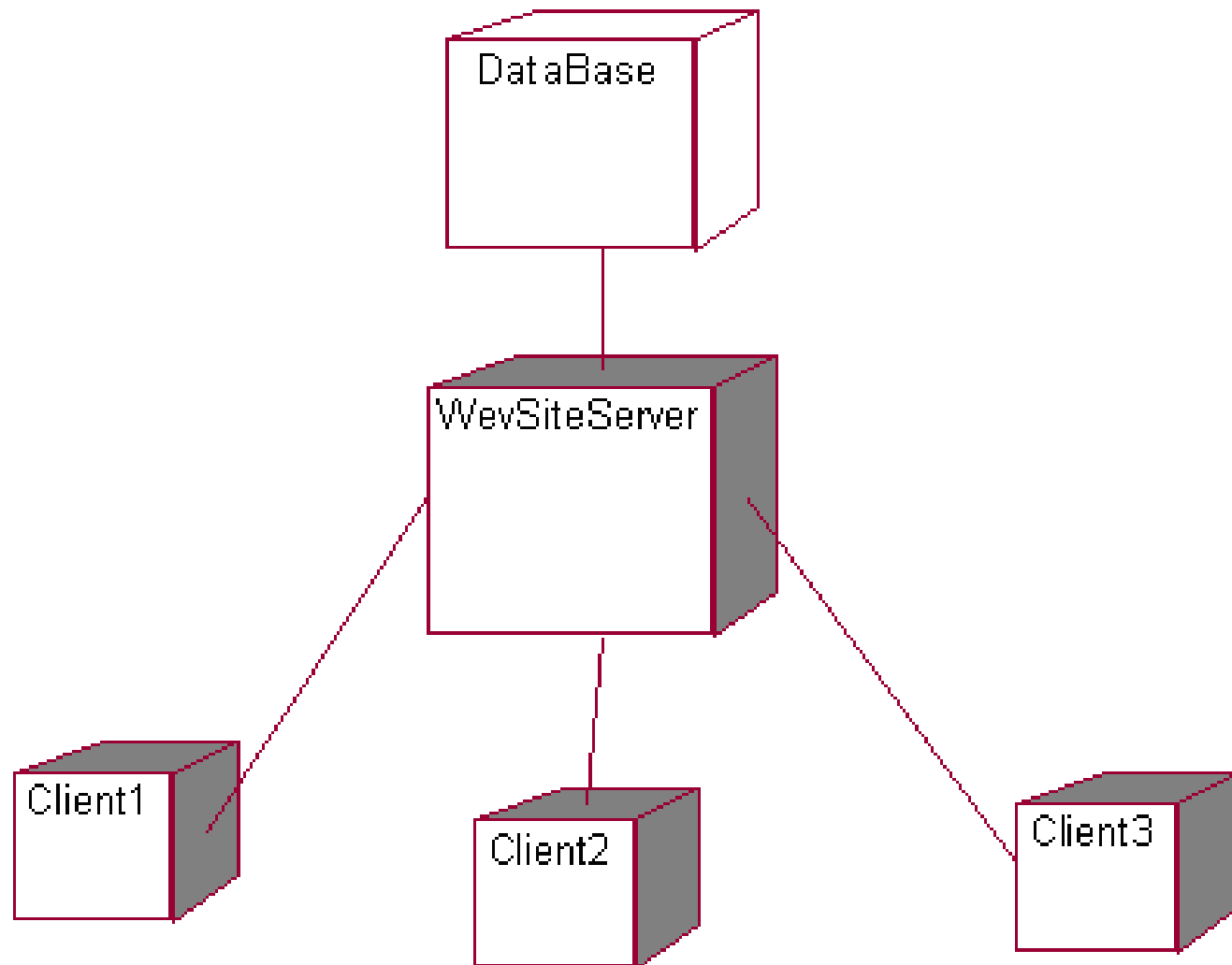
- 网络教学系统的组件图如下图，组成Web应用程序的页面包括：维护页面（Maintenance Page）、文件下载页面（File Download Page）、文件上传页面（FileUpload Page）、信息发布页面（Message Issue Page）和登录页面（Login Page）





## ●9、系统的配置图

- 配置图主要是用来说明如何配置系统的软件和硬件。网络教学系统的应用服务器负责保存整个Web应用程序，数据库是负责数据库管理。此外还有很多终端可以作为系统的客户端。由于客户端很多，在此只画出3个客户端，系统配置图如下图



Thank you !