



中国海洋大学信息学院计算机科学与技术系

软件工程

第三章 需求分析 (part2)

2020 / 11



内容提要

01 ER图与用例图

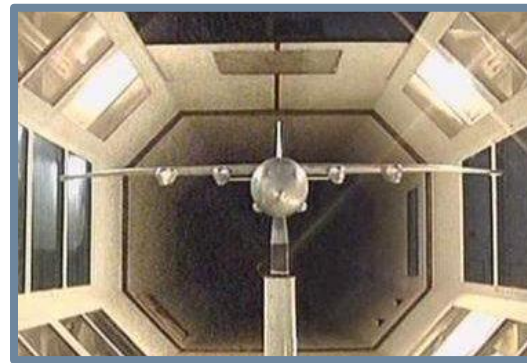
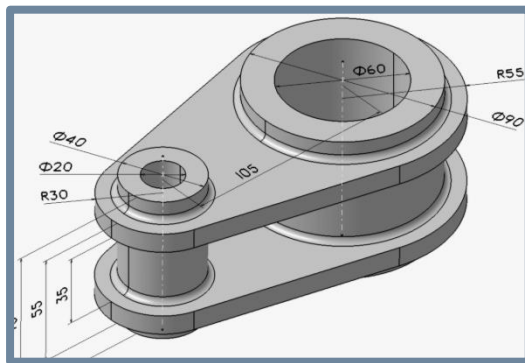
02 用例建模的主要步骤

03 用例建模的主要工具



工程学科的重要特征是

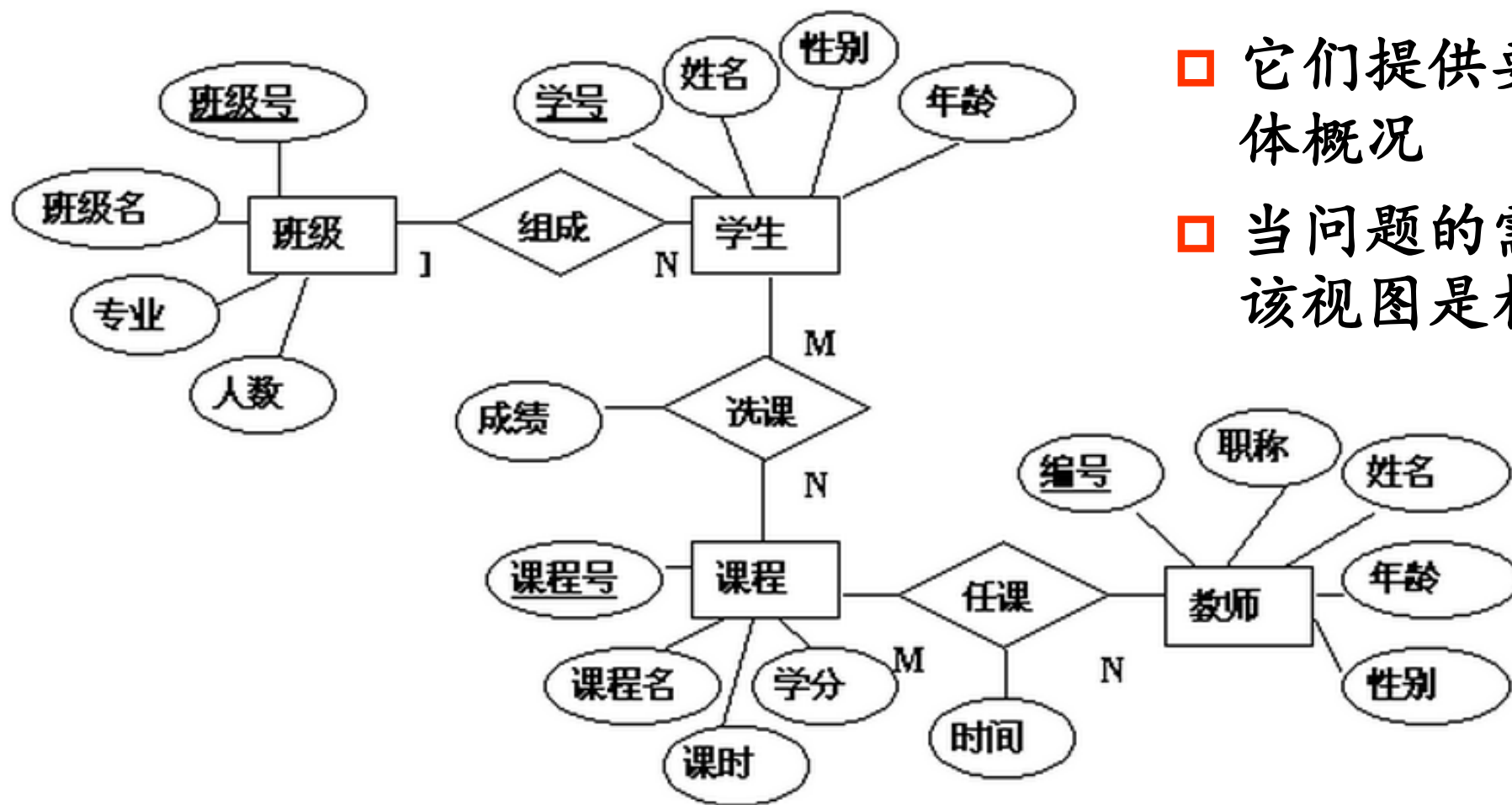
- 过程可重复
- 有标准的符号用于建模、编档、交流



非常好的建模工具 —— ER图

- 用来表示概念模型
- 有三个核心结构
 - 实体：用矩形表示，代表真实世界的对象，有属性和行为
 - 关系：两个实体间的边，中间有一个菱形
 - 属性：表示和实体有关的数据或属性
- 常用于需求分析的早期建模

非常好的建模工具 —— ER图



ER图被广泛应用，是因为：

- 它们提供要解决问题的总体概况
- 当问题的需求发生变化时，该视图是相对稳定的

用例图

ER图只给出问题的全貌，是结构化、静态的，没有交互

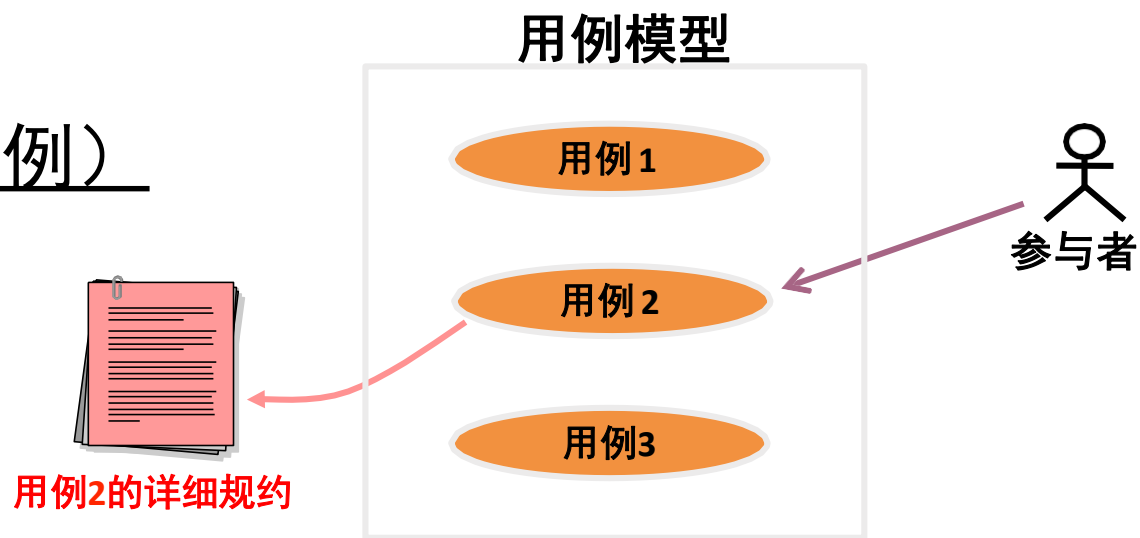
为进一步补充，我们需要使用用例图

■ 构成

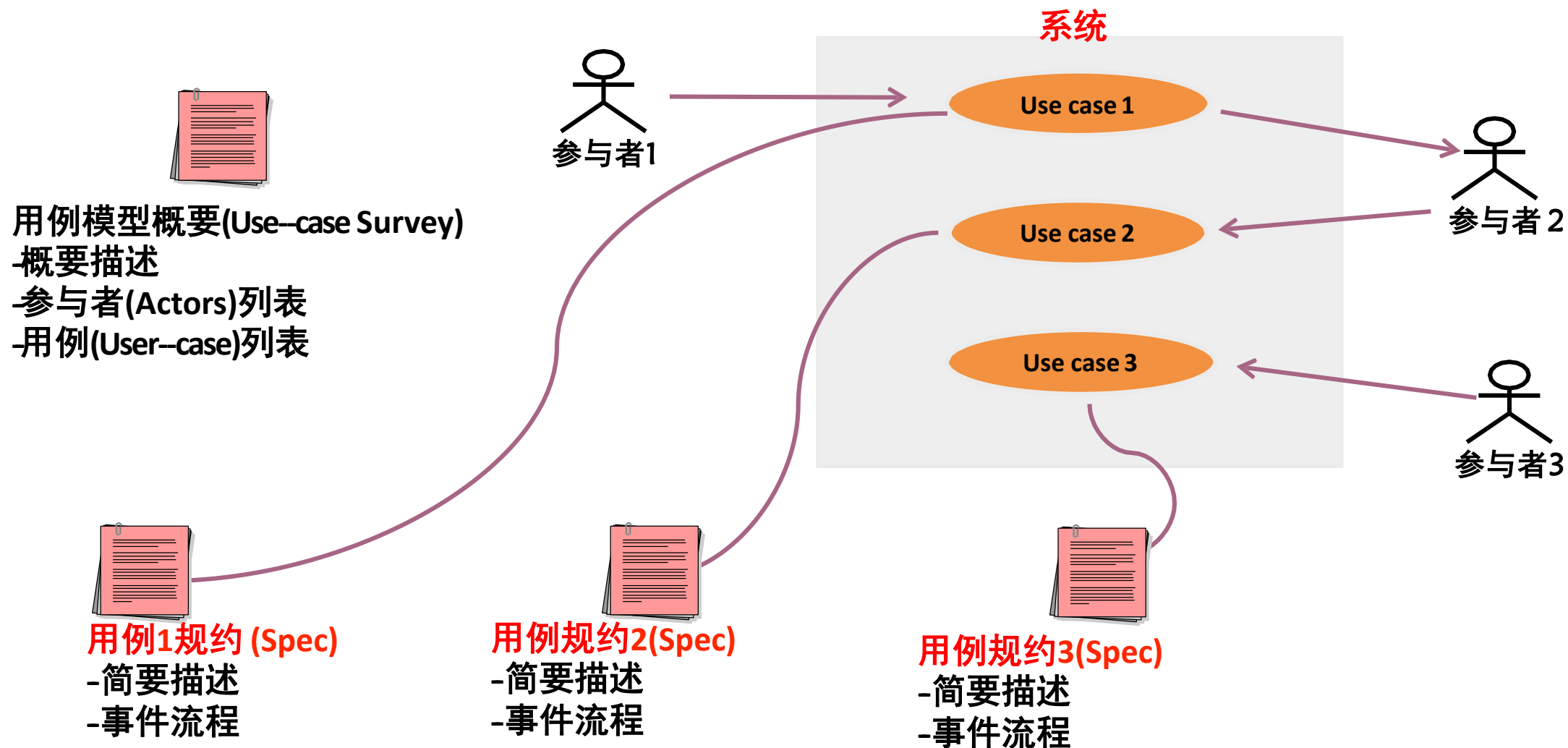
- 大方框: 系统边界
- 火柴人: actors, 人或系统
- 椭圆: 主要的功能
- 线: 连接actor和use case

用例建模——描述系统的功能性需求

- 关联于系人需要以及软件需求
- 确认与系统交互的人或对象（参与者）
- 定义系统的边界
- 捕捉和传达系统的理想行为（用例）

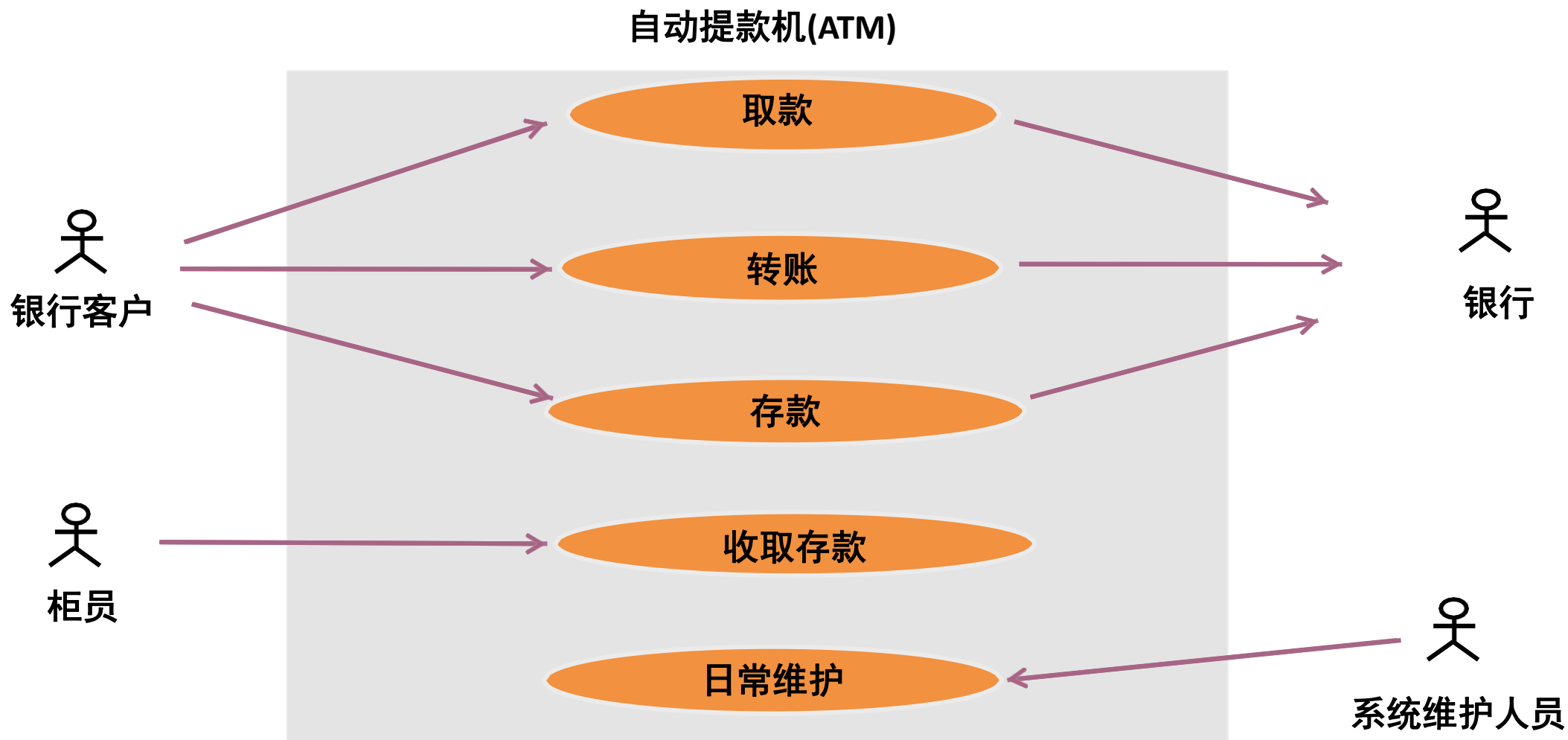


用例模型的表示——文本描述

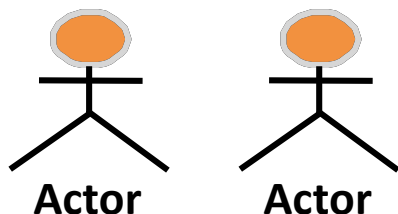


我们将用ATM机的例子，让大家对于用例图有一个直观的认识

用例模型的表示——用例图



用例图的主要元素



参与者 (Actor)

与系统交互的人或外部系统



用例 (Use case)

系统为参与者提供的有价值的服务功能



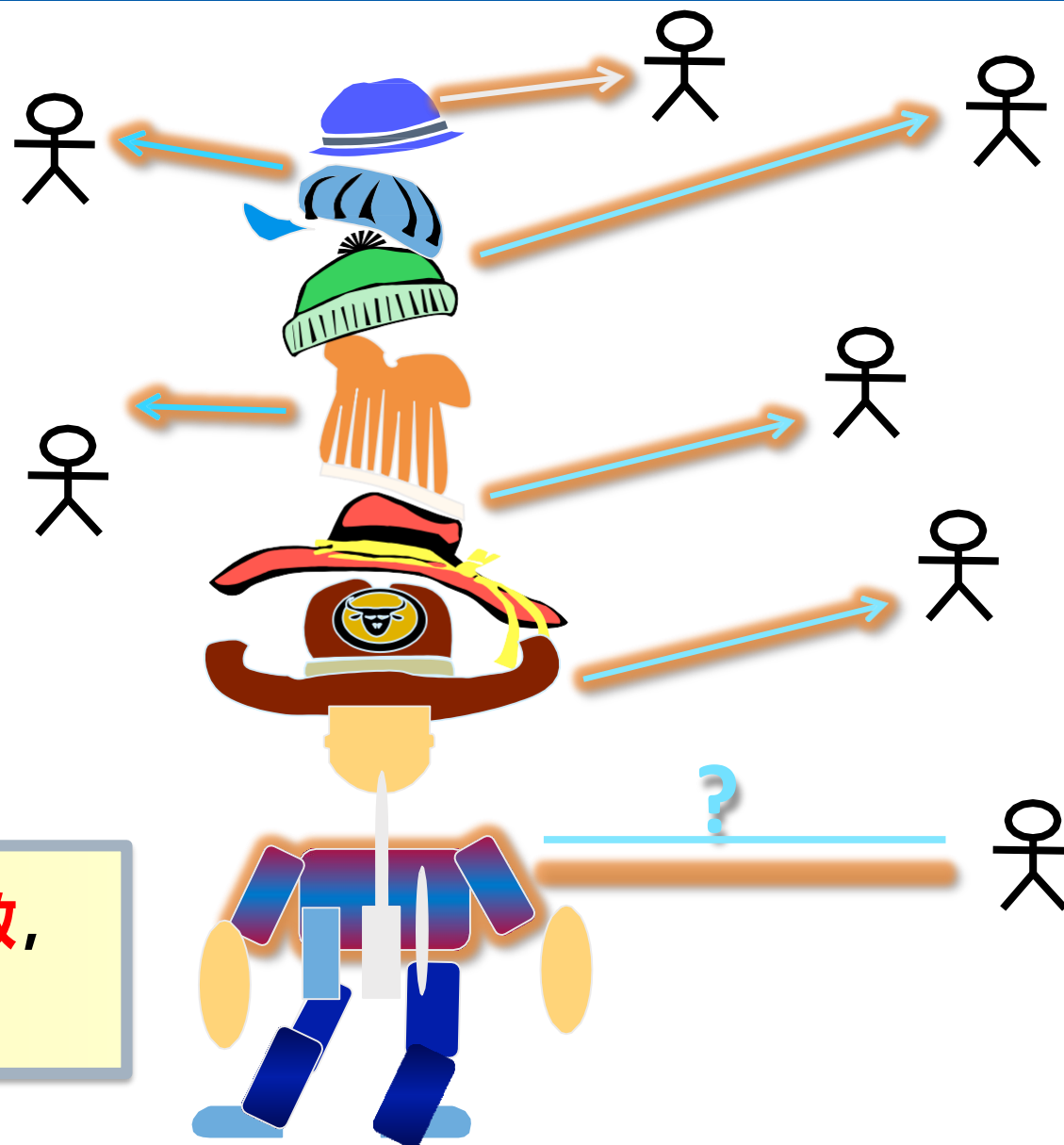
关联 (Association)

用例图中用例与参与者之间的交互关系

参与者的定义：关注角色

- 与系统交互的人
- 与系统交互的硬件组件
- 或者其他的外部系统
- 关注的重点是所承担的“角色”
- 参与者命名要明确其角色

学生有可能是选课用户，也有可能是助教，
所以应该注意承担的“角色”



什么是用例？



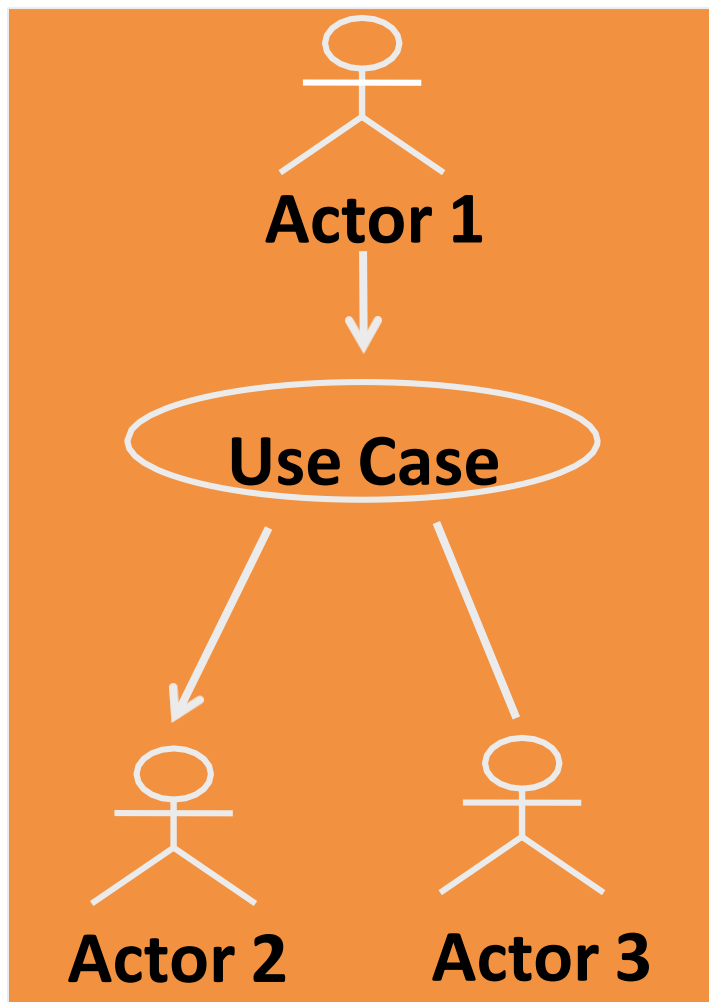
一个用例

定义**系统**的一系列**行为**

通过此可为**参与者**提供**有价值且可观测**的结果

注意：命名为“用户注册”而且不是“注册用户”

交互——关联 (Association)



- 参与者与用例之间的交互通道
- 用一条直线表示交互——关联
 - 有箭头的关联指出是谁发起的交互
 - 没有箭头则表明双方都可以发起交互

内容提要

01 ER图与用例图

02 用例建模的主要步骤

03 用例建模的主要工具



构建用例模型的步骤

第一步：找到所有的参与者和用例

- 识别出参与者并做简单的描述
- 识别出用例并做简单的介绍

第二步：编写用例

- 列出用例
- 给用例事件流程划分重要等级
- 按照重要程度排序详细描述事件流程



寻找参与者

- 谁/什么使用系统？
- 谁/什么从系统中获取信息？
- 谁/什么向系统提供信息？
- 公司的哪个部门会使用系统？
- 谁/什么负责系统的维护？
- 还有哪些其他系统会使用系统？



通过这些问题，可以帮助我们找到参与者

参与者的描述

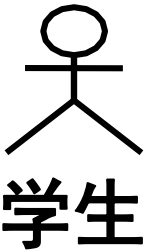
名称

学生

简要描述

注册课程的用户

和用例之间的关系



用例描述

用例的描述

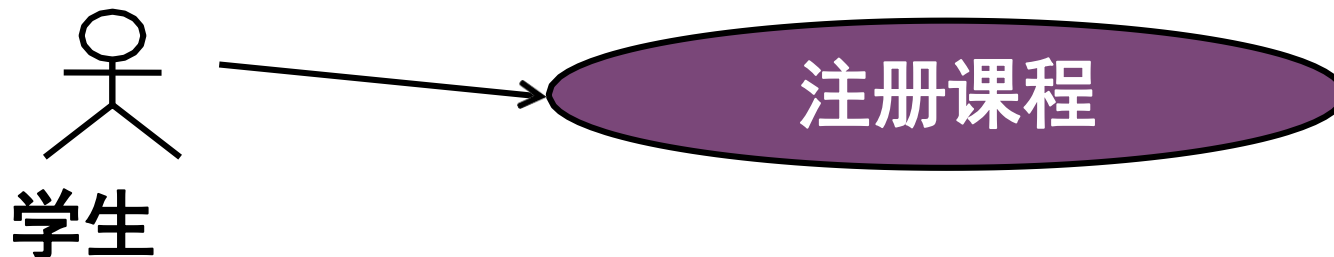
名称

注册课程

简要描述

学生选择下个学期想上的课程。生成必修课和选修课的课表信息。

与参与者的关系



构建用例模型的步骤

第一步：找到所有的参与者和用例

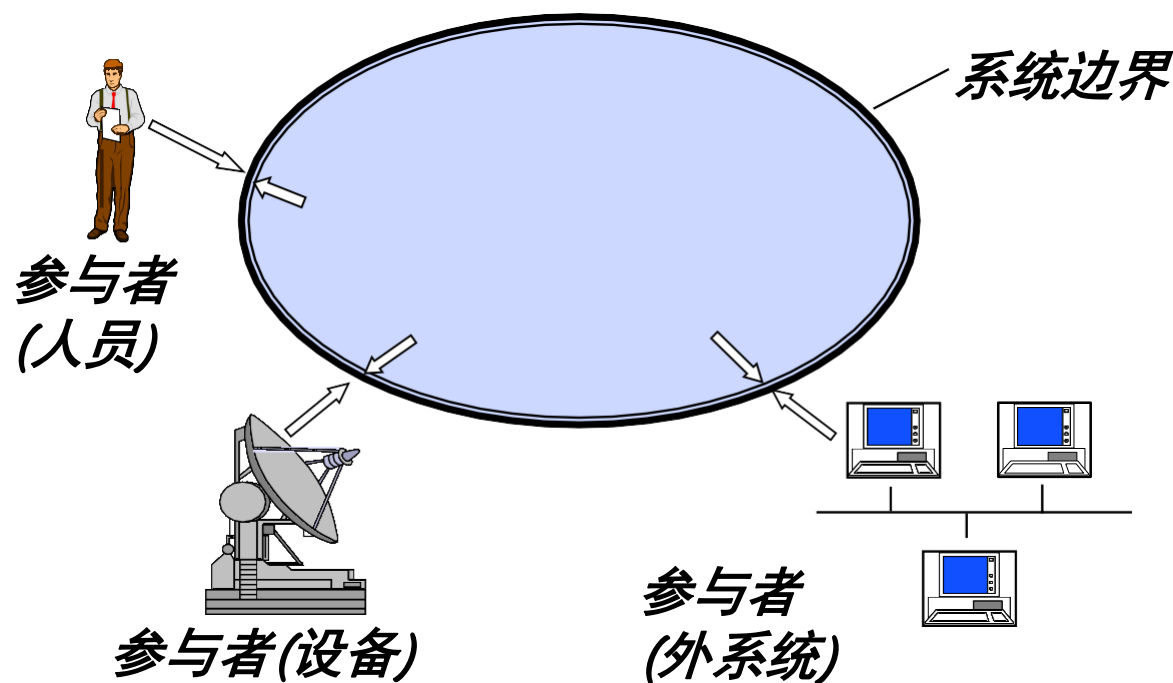
- 识别出参与者并做简单的描述
- 识别出用例并做简单的介绍

第二步：编写用例

- 列出用例
- 给用例事件流程划分重要等级
- 按照重要程度排序详细描述事件流程

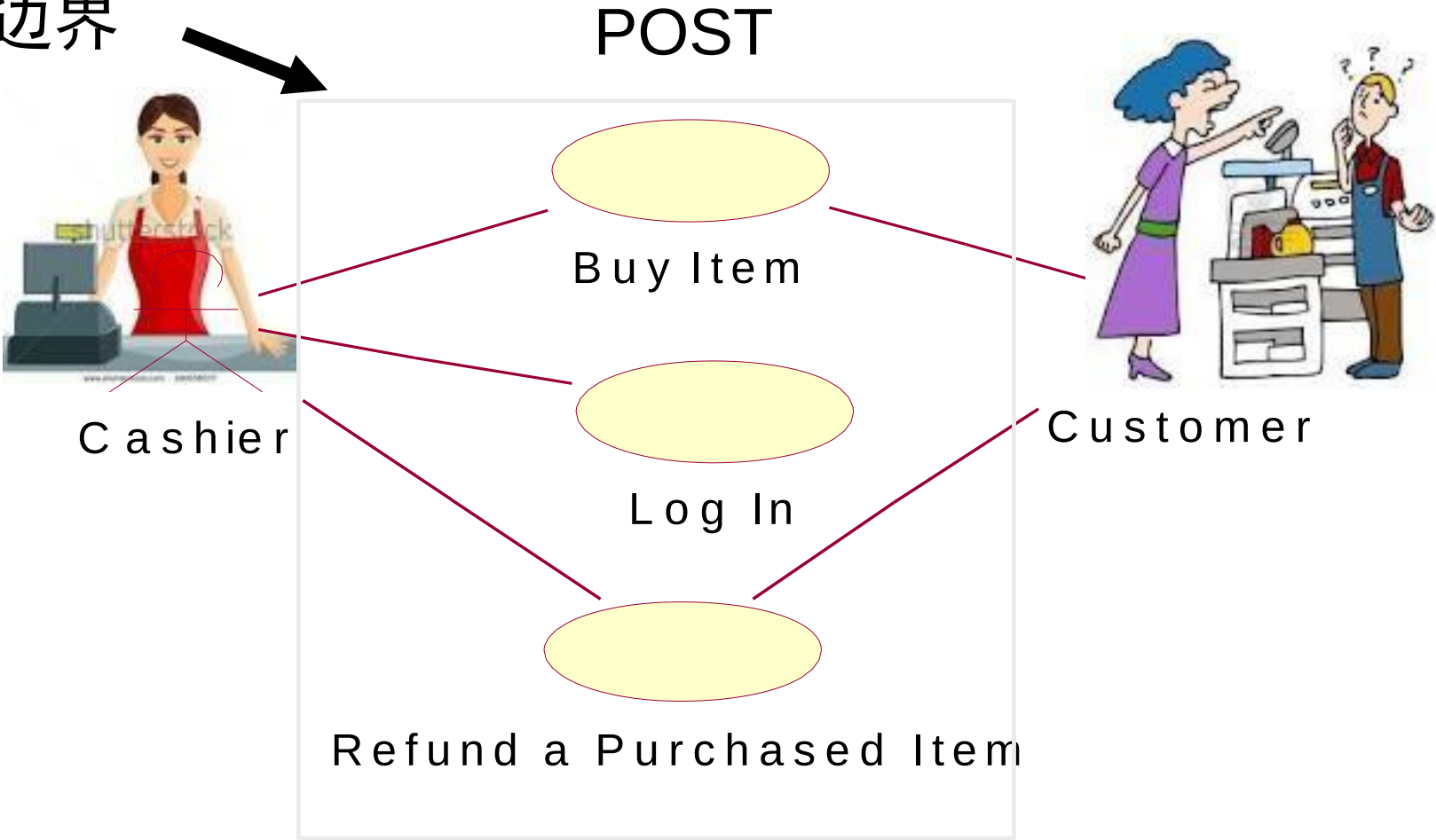


- **系统边界:** 一个系统所包含的所有系统成分与系统以外各种事物的分界线
- 系统边界会对用例以及Actor的定义有所影响



(本图选自北大麻志毅老师教材)

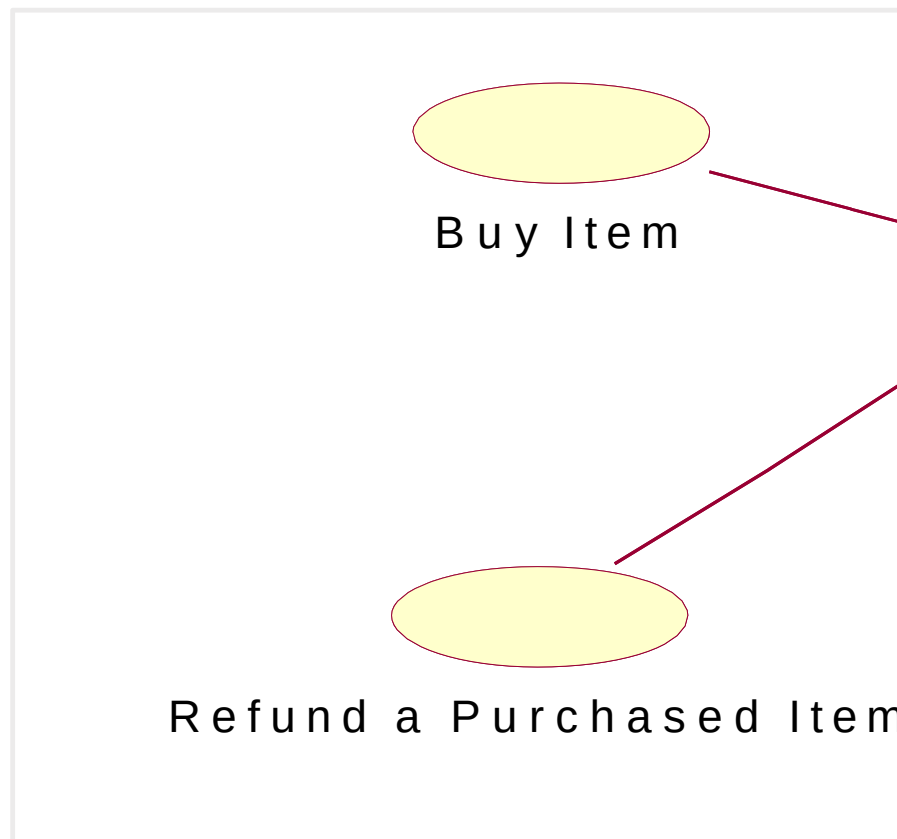
系统边界



系统边界

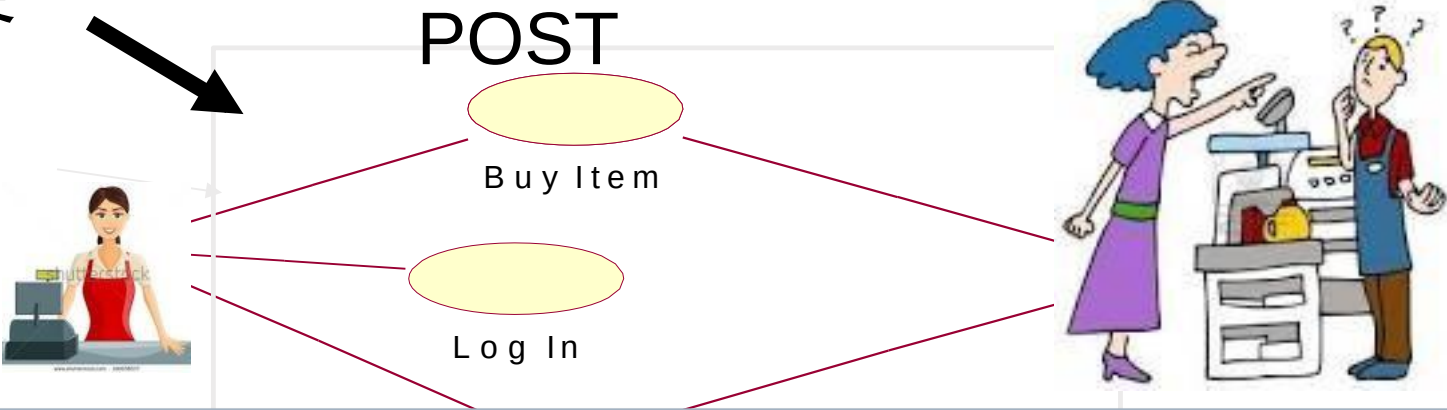


POST



Customer

系统边界



不同系统边界定义决定了与系统交互的对象，
参与者与系统交互的方式，进而影响用例模型的设计



Administrator

Manage Users

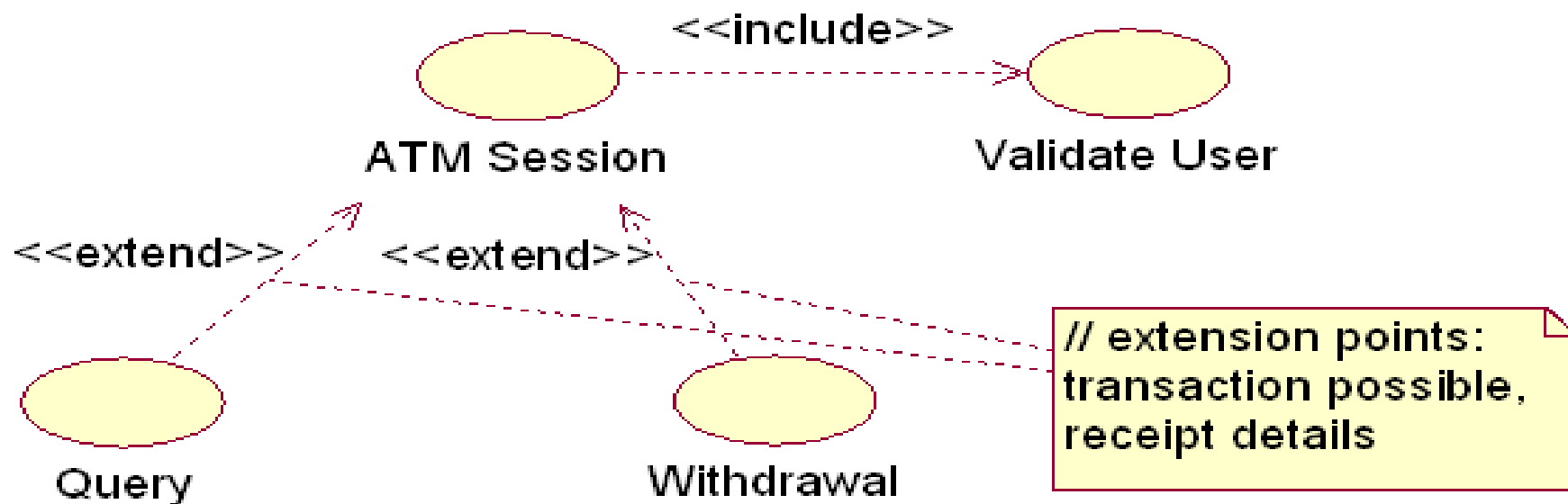
Add a Lot more...



Manager

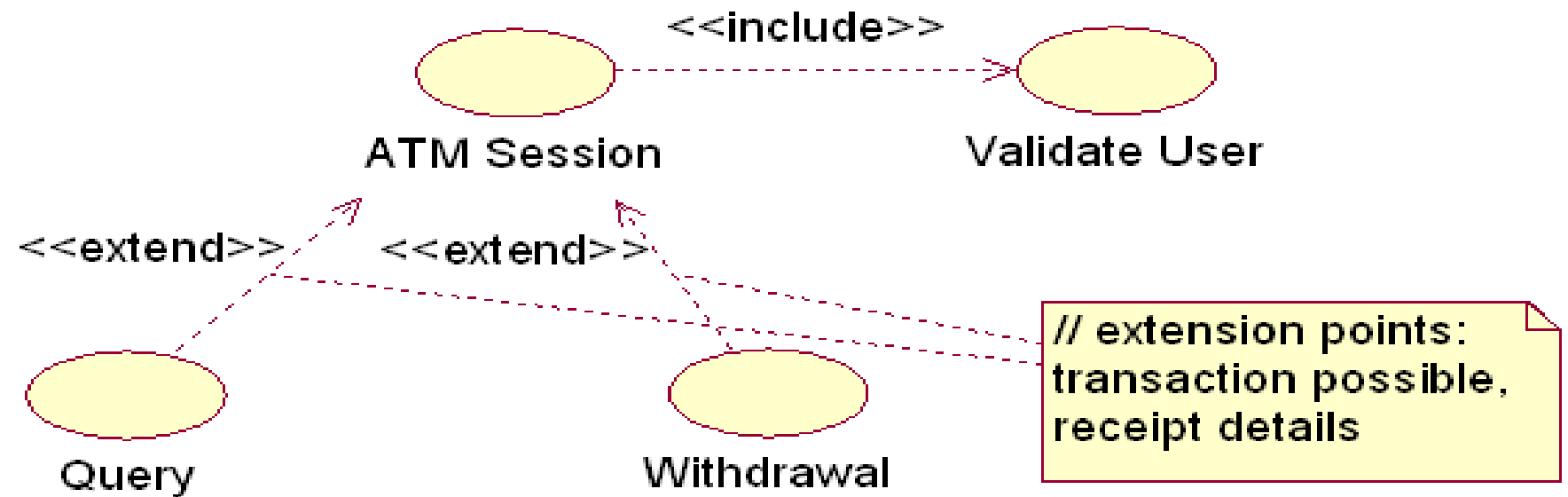
用例间的包含关系

- 当多个用例有共享行为时，使用包含关系
- 为共享行为单独创建用例，被相关用例“包含”
- 例如：
 - 估价
 - 权限检查

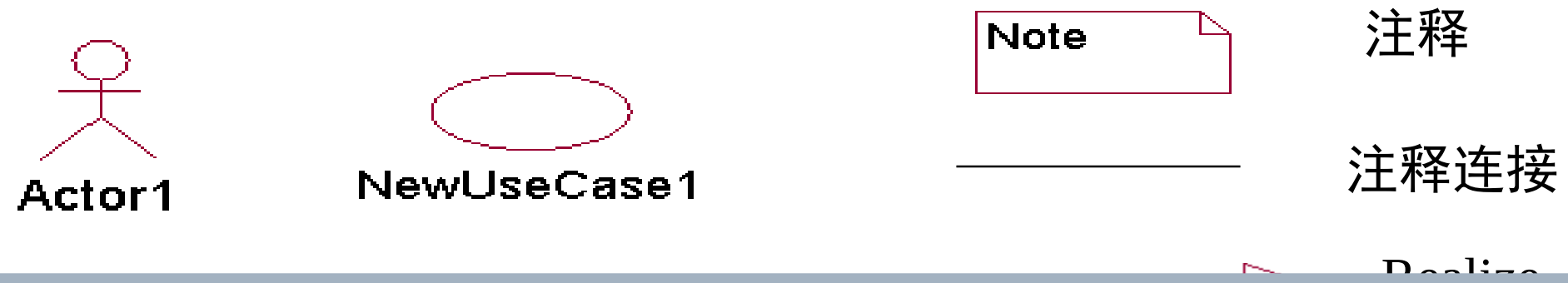


用例间的扩展关系

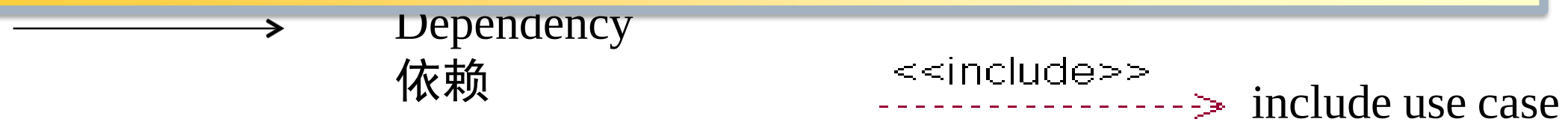
- 一个用例与另外一个用例近似，只有少许额外的活动
- 将代表普遍或基本行为的情况定义为一个用例
- 将特殊的、例外的部分定义为扩展用例



用例图中的主要图标



对于用例建模而言，没有绝对正确的答案，我们要做的，
就是符合一定的建模风格，把用户的需求表达清楚



说明：UML中不使用颜色来作为图形语义的区分标记

内容提要

01 ER图与用例图

02 用例建模的主要步骤

03 用例建模的主要工具

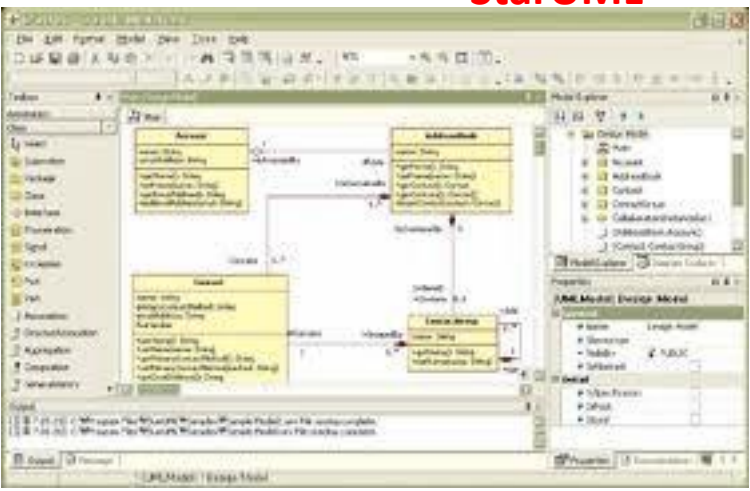


建模工具的主要功能

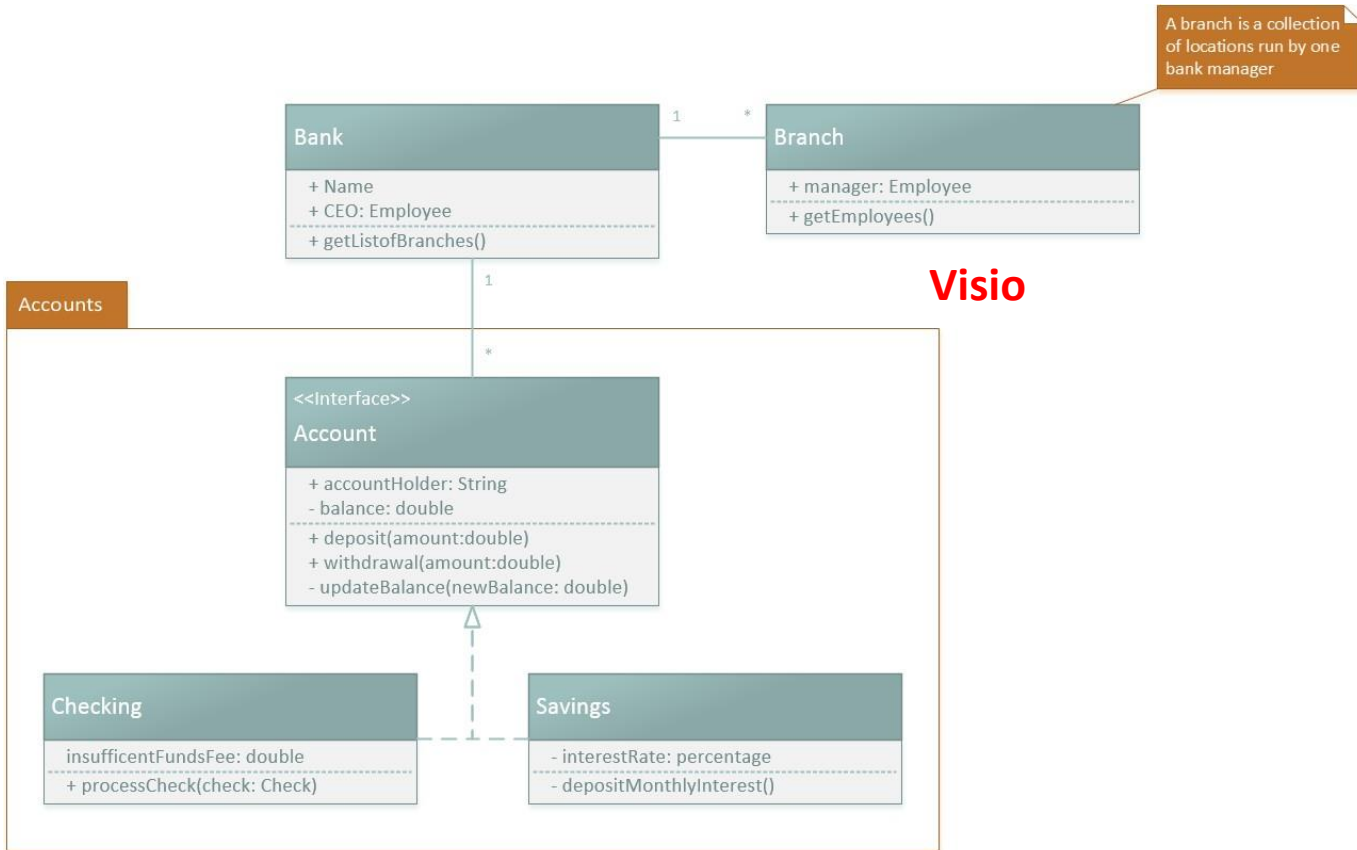
- 可视化模型表达
 - UML模型
 - Web模型，例如Azure
 - 数据库模型，例如Power Designer
 - 用户自定义模型，例如Visio

- 画图工具

StarUML



- 辅助开发流程中的项目管理



常用的建模工具

- Caliber (version 11.4.2) , Borland (Micro Focus)
 - 应用于需求管理、建模可视化
- inteGREATE (version 8.7.13), dDev Technologies
 - 应用于需求管理, 需求开发, 建模可视化
- VersionOne (version 15.0.9), VersionOne
 - 应用于需求管理, 敏捷开发
- StarUML (version 5.6.5), MKLab
 - 应用于UML建模

资源链接: http://en.wikipedia.org/wiki/List_of_Unified_Modeling_Language_tools

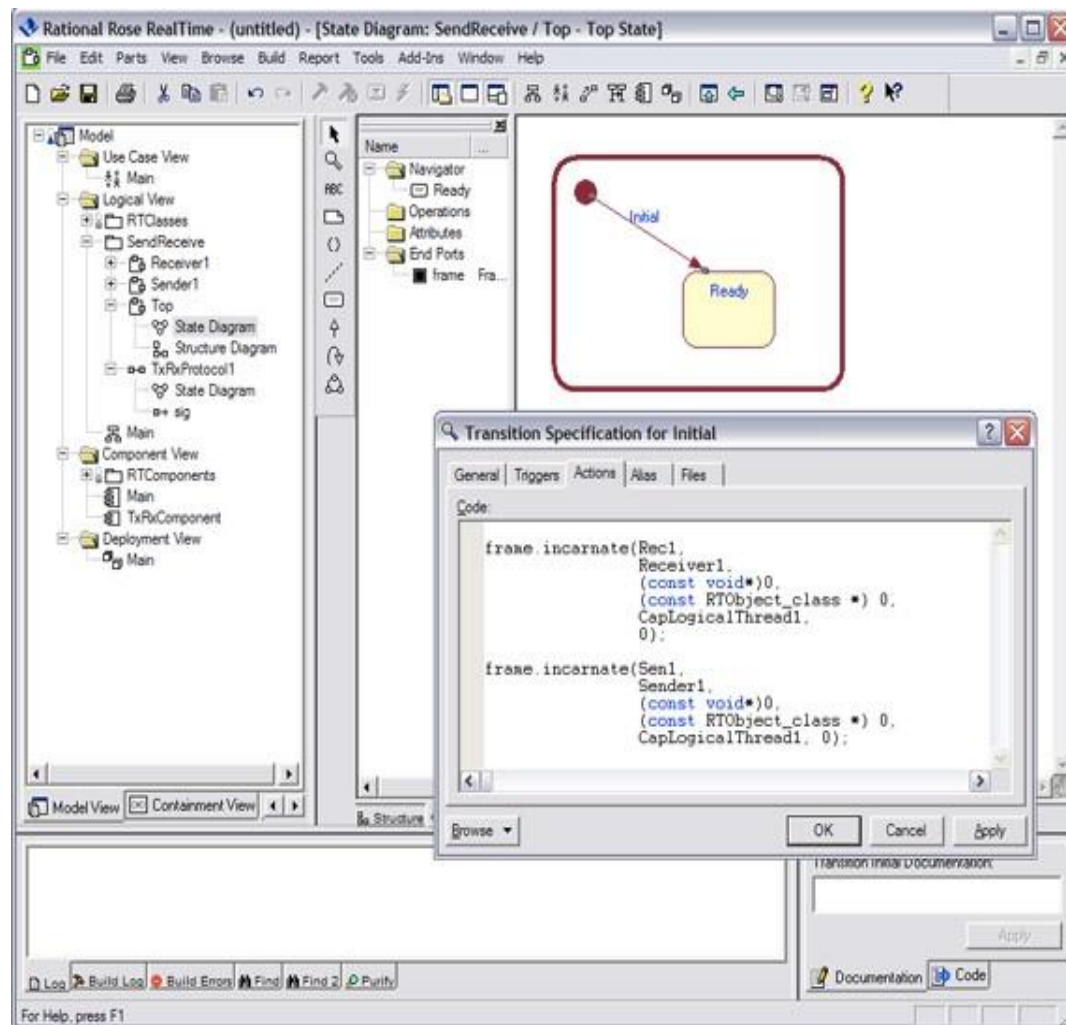
资源链接: <http://makingofsoftware.com/resources/list-of-rm-tools>

常用的建模工具 (UML 2.0)

- IBM Rational Rose
 - Rational是IBM设计的集成设计、建模和开发软件应用程序套件。Rose是其中的可视化建模工具
 - 适用于应用程序开发、数据建模、Web Service设计、业务建模、组件建模等
 - 支持模型与代码之间的转化，逆向工程

资源链接:

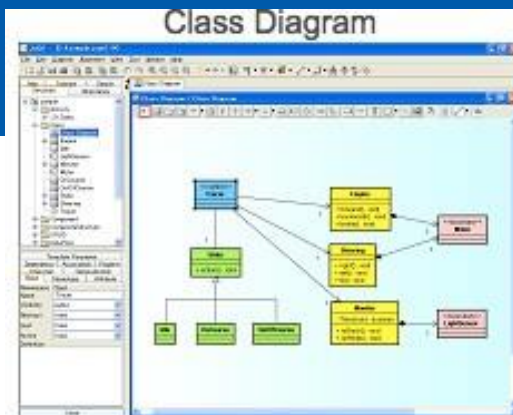
<http://www.ibm.com/developerworks/cn/rational/uml>



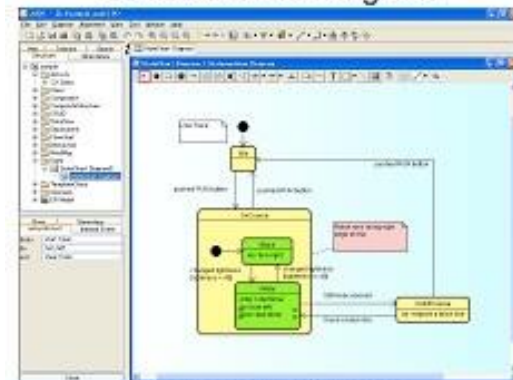
常用的建模工具 (UML 2.0)

- JUDE

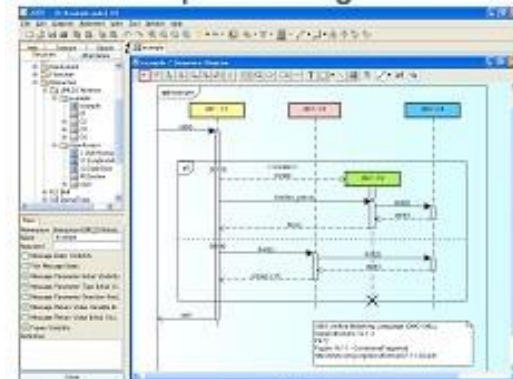
- 轻量级的UML建模工具
- 支持项目团队中的协同工作
- 文档转换
- 全面支持UML1.4，支持UML2.0中部分模型



Class Diagram



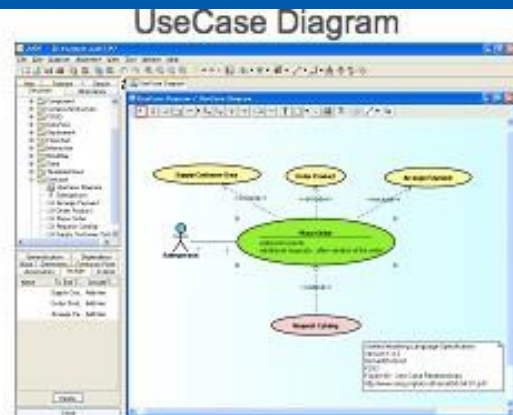
State Machine Diagram



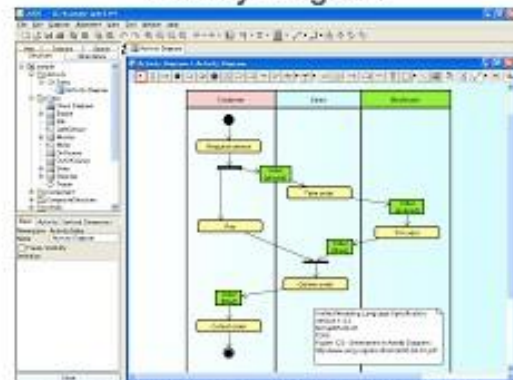
Sequence Diagram



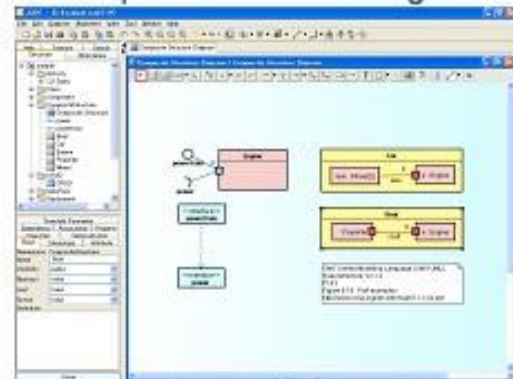
Component Diagram



Use Case Diagram



Activity Diagram



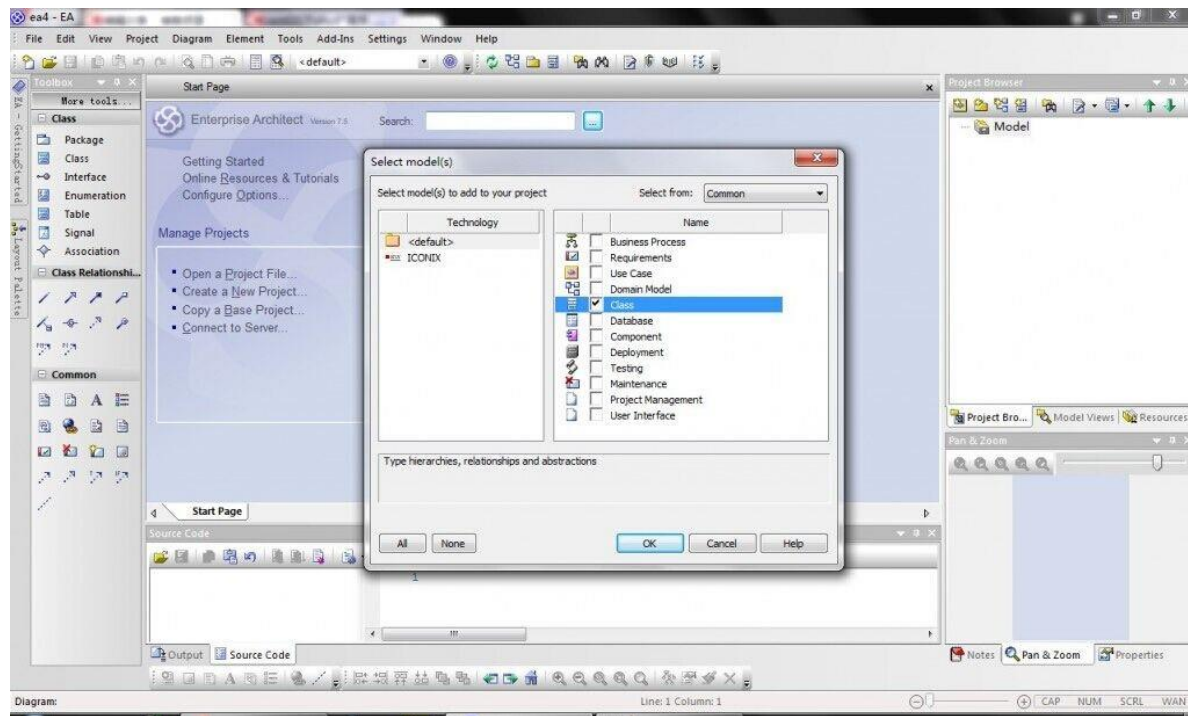
Composite Structure Diagram



Deployment Diagram

常用的建模工具 (UML 2.0)

- Enterprise Architect (EA)
 - 全功能的，基于UML2.0的可视化工具
 - 用于设计、编写、构建并管理以目标为导向的软件系统。
 - 为整个团队提供系统开发不同阶段所需的信息建模，服务于不同身份的成员
 - 提供从需求分析、软件设计一直到执行和部署整个过程的全面可跟踪性
 - 支持多种语言的前向或逆向生成代码工程



聆听

总结

分享

改进