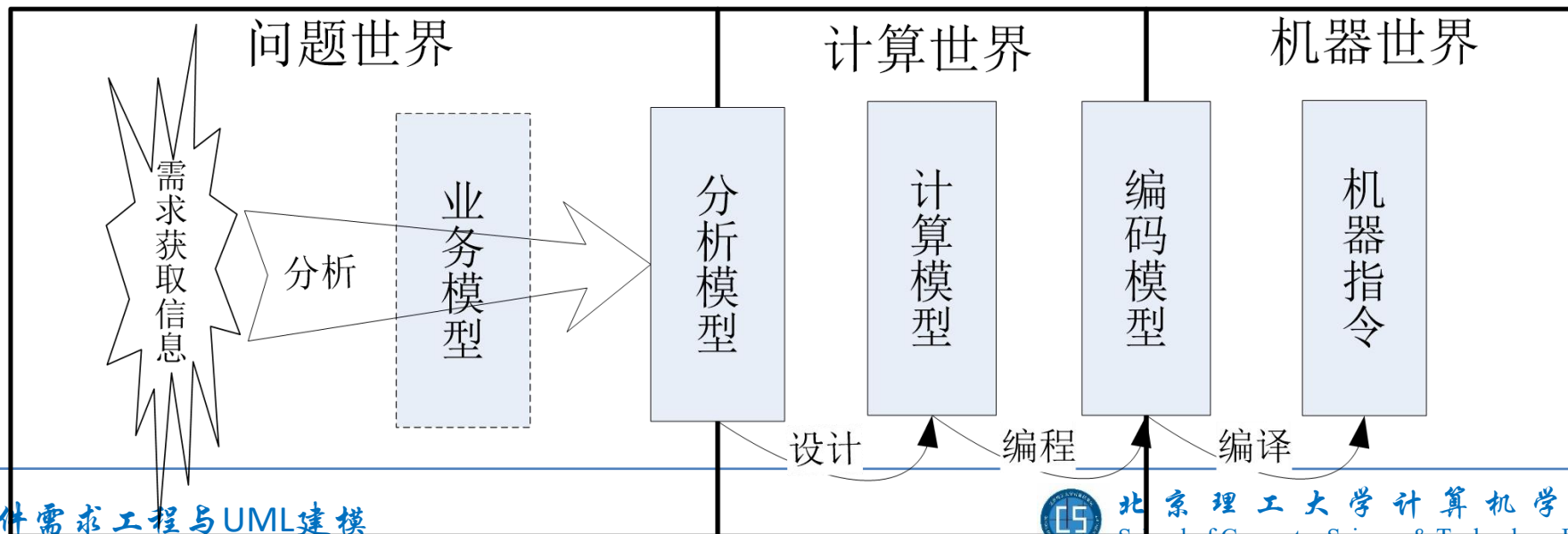
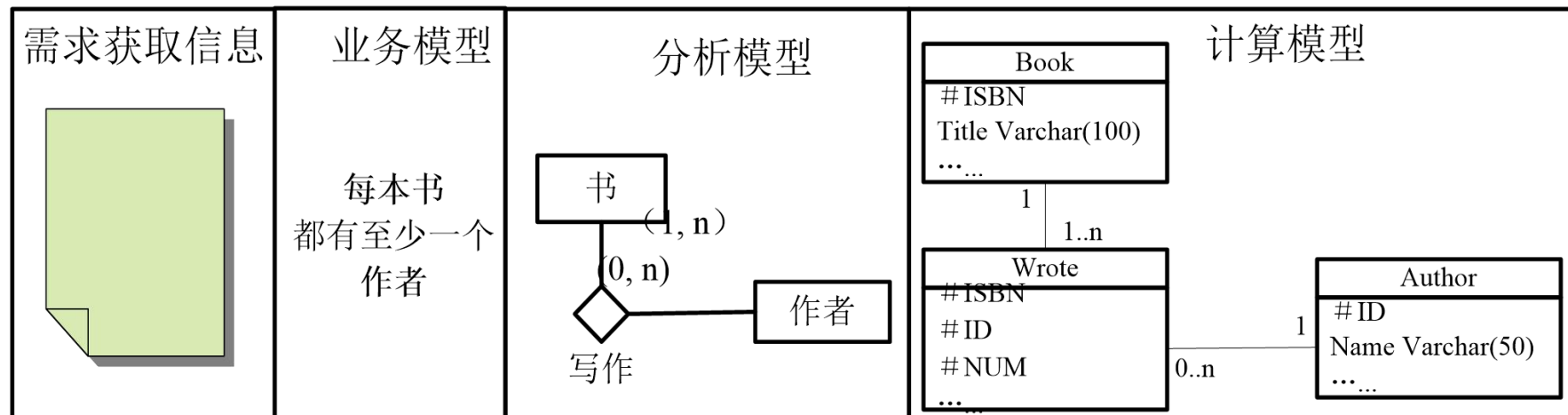




第6章、第7章回顾

1.1. 建立需求分析模型

■ 三种模型



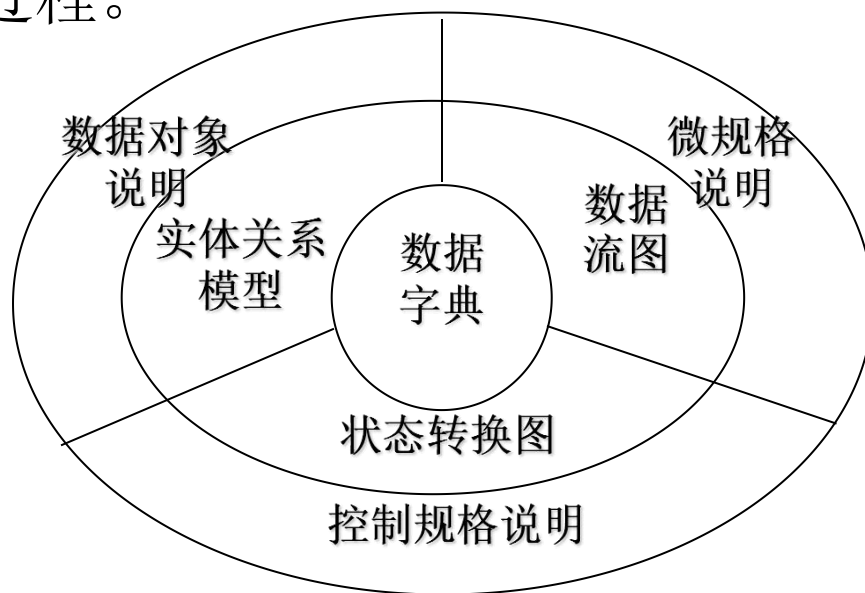
软件需求工程与UML建模



结构化需求分析与建模

结构化分析概述

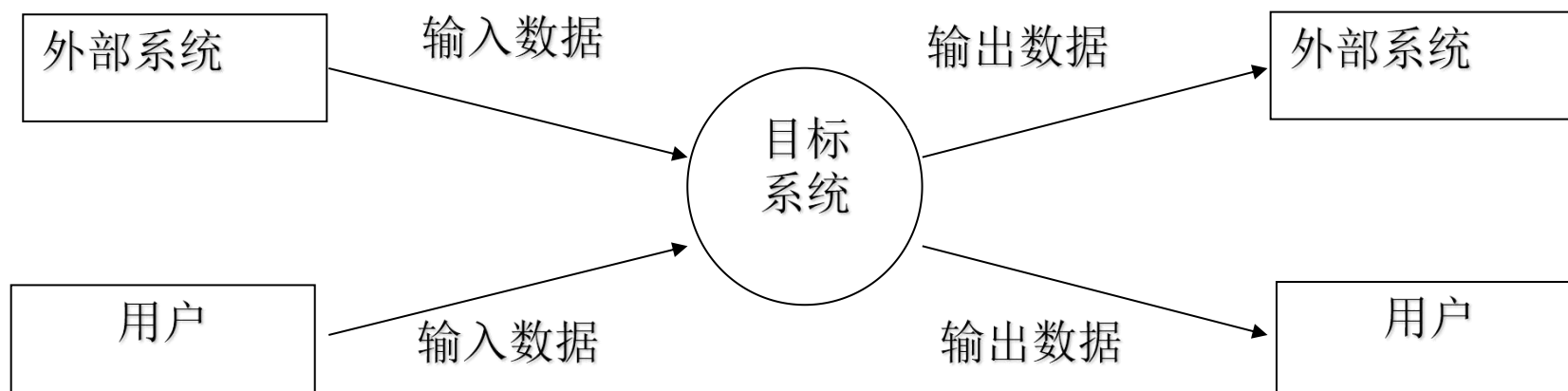
结构化分析的核心是数据。数据包括在分析、设计和实现中涉及的概念、术语、属性等所有内容，并把这些内容定义在数据字典中。围绕数据字典，完成功能/过程模型、数据模型和行为模型的结构化建模过程。



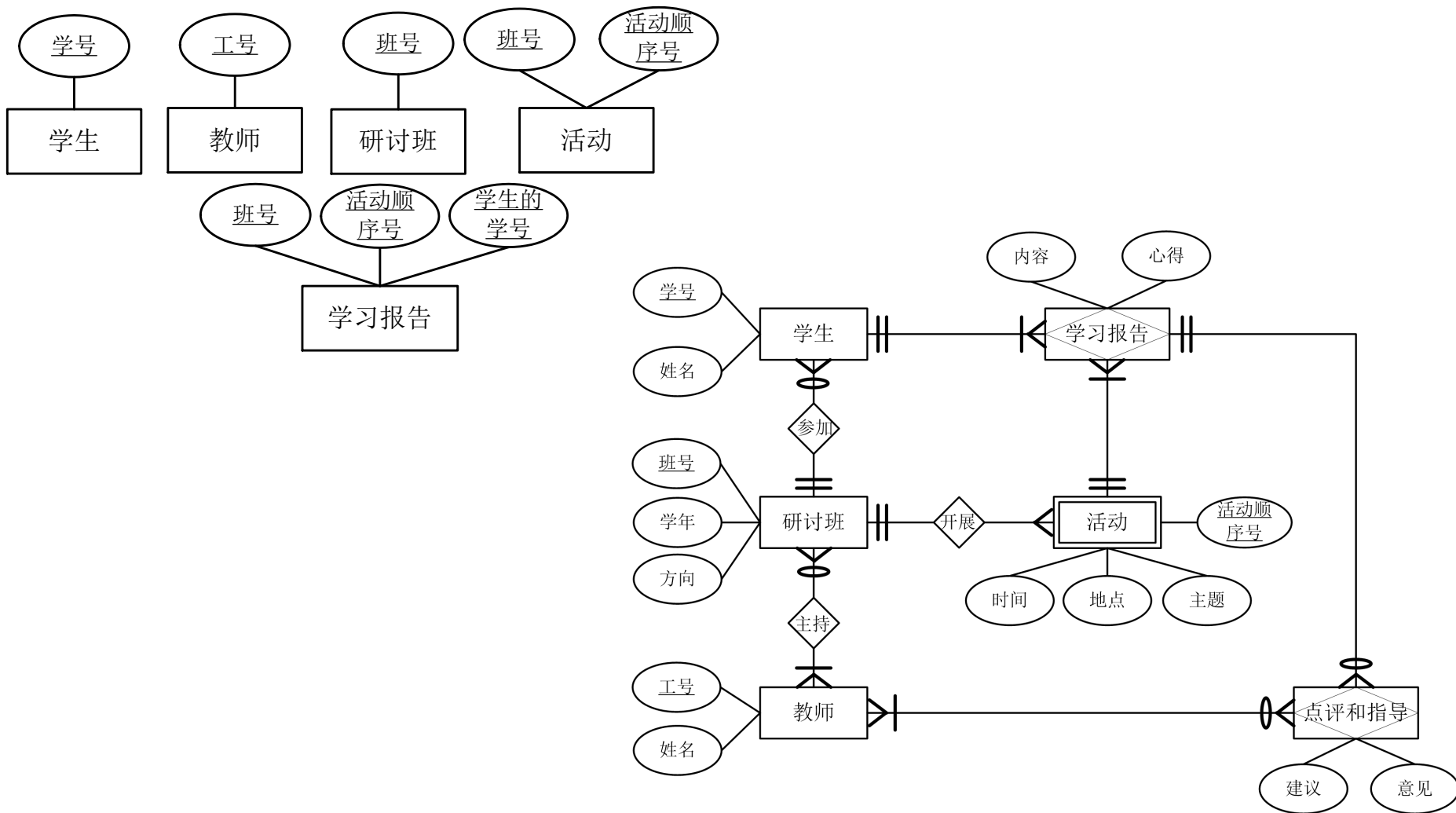
1.1. 过程建模：数据流图

面向数据流的过程建模

数据流图（Data Flowing Diagram, DFD）是结构化建模中最流行的功能建模工具。DFD描述从数据输入、数据转换到数据输出的全过程。能对DFD图分层，分层的DFD更进一步刻画了系统的功能分解。



1.2. 数据建模: 实体关系图示例



第8章.面向对象分析与 UML建模

韩锐

北京理工大学 计算机学院

Email: 379068433@qq.com
hanrui@bit.edu.cn

面向对象分析与UML建模(1)

- 面向对象方法概述
- 面向对象分析 VS 结构化分析
- UML模型概述
- UML和Rose

现实世界的复杂模型

- 复杂总是简单部分的组合
- 简单部分又是更简单部分的组合
 - 简单组成复杂的过程存在层次性
- 每个最小简单部分独立负责完成一系列相关任务
 - 低耦合高聚合：相比较而言，每个组合内部各部分的关系比其内部与外部的关系都更紧密
- 各个部分通过一致的接口进行组合，即一个部分对其它部分的所知仅仅是接口

映射现实模型的面向对象思想

- 任何系统都是能够完成一系列相关目标和任务的对象
 - ▣ 对象完成一个任务时会请求一系列其他对象帮助其完成一些子目标
 - ▣ 其他对象为了完成其任务又会请求将子目标更细分为子子目标，并请求其他对象帮助完成
- 子目标的分解和责任分担一直进行直到最后产生的子部分可以映射到计算实体
 - ▣ 计算实体：对象
 - ▣ 层次关系：聚合（组合）、继承、关联
 - ▣ 组合接口：一个对象暴露的接口

面向对象分析基本思想

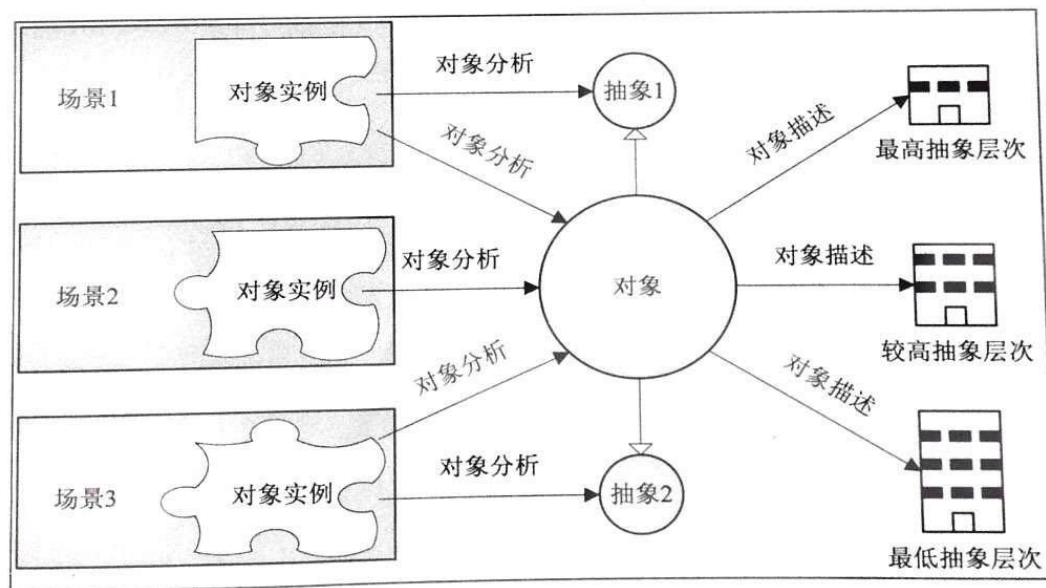
■ 思想方法

- 从现实世界中客观存在的事物出发，直接以问题域中的事物为中心，思考 and 认识问题。
- 将事物的本质特征和系统责任，抽象表示为系统的对象，作为系统的基本构成单位，建立软件系统。
- 强调运用日常逻辑思维经常采用的思想方法和原则，例如：抽象、分类、继承、聚合、封装和关联，并以易懂方式表达。



面向对象分析方法

- 一切都是对象
- 对象都是独立的
- 对象具有原子性
- 对象都是可抽象的
- 对象都有层次性



面向对象方法基本概念

- (1)类和对象
- (2) 封装和消息
- (3) 继承
- (4) 多态性



(1) 类和对象的概念

- **类**是抽象及分类的概念，是人们分析问题的基本方法，分类是划分共性的事物为一类。抽象是忽略事物非本质的特征；
 - **数据抽象**是根据施加在数据上的操作来定义数据的类型；
 - **过程抽象**是将确定的功能定义为单一实体（确定的功能是由一系列操作完成的）；
- 在外部世界中，相同属性和操作的对象属于一个类，**对象**是类的一个实例；
 - 在机器空间中，类是一个可复用的模板，而对象是复用出来的独立可执行程序块



例子

■ 对象包括实体对象和无形对象

数据实体抽象：学生、客户、电子邮件、汽车
无形过程抽象：轨迹、交易、答辩、课程



例子

例：考虑下列事物的分类：

显微镜、摩天楼、望远镜、飞机、铆钉、管道、卡车、单向阀、指数、过滤器、压力器、自行车、正旋、岩洞、眼镜、车库、滑翔机、平方根、铁钉、水龙头、螺钉、余旋、螺栓、帐篷、双筒镜、车棚、杂物堆房、摩托车、帆船、瞄准器、

分类：

光学设备类：显微镜、眼镜、望远镜、瞄准器、双筒镜。

管道控制类：管道、单向伐、水龙头、过滤器、压力器。

交通工具类：自行车、帆船、卡车、飞机、滑翔机、摩托车。

五金类：铁钉、螺钉、螺栓、铆钉。

建筑类：帐篷、岩洞、车棚、车库、杂物堆房、摩天楼。

数学概念类：平方根、指数、正旋、余旋。



例子

抽取分类后事物的属性和特征：

光学设备类： 聚焦度、变焦长度、单双镜形式、用途功能

交通工具类： 时速、功率，动力资源、消耗、用途、

五金类： 材料、直径、长度、硬度、用途

建筑类： 面积、高度、材料、用途

数学概念类： 概念定义、公式内容、用途



(2) 封装和消息的概念

- **封装**：把类的内部属性和一些操作隐藏起来，只将公共的操作对外可见。避免外界错误和内部修改带来的影响；
 - 对象只通过消息来请求其他的对象执行自身的操作；
- **消息**必须直接发给指定的对象，消息中包括请求执行操作的必要信息；
 - 一个对象是消息的接收执行者，也可以是消息的请求发送者。

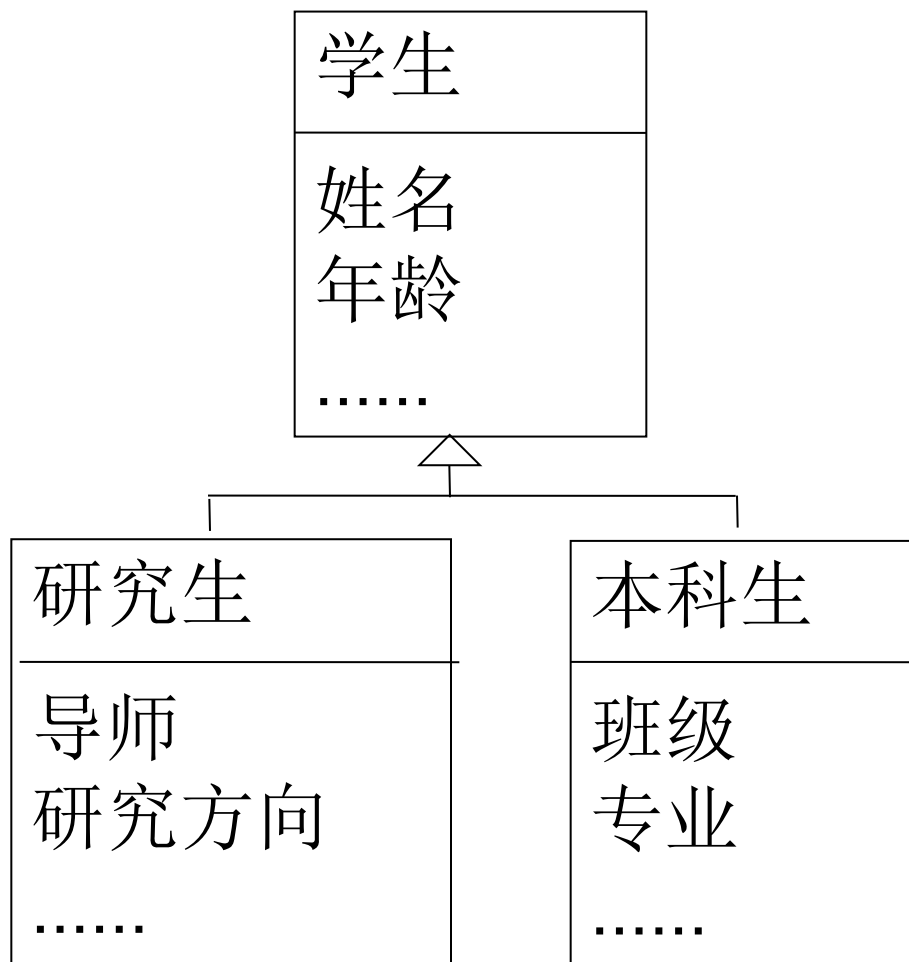


(3) 继承的概念

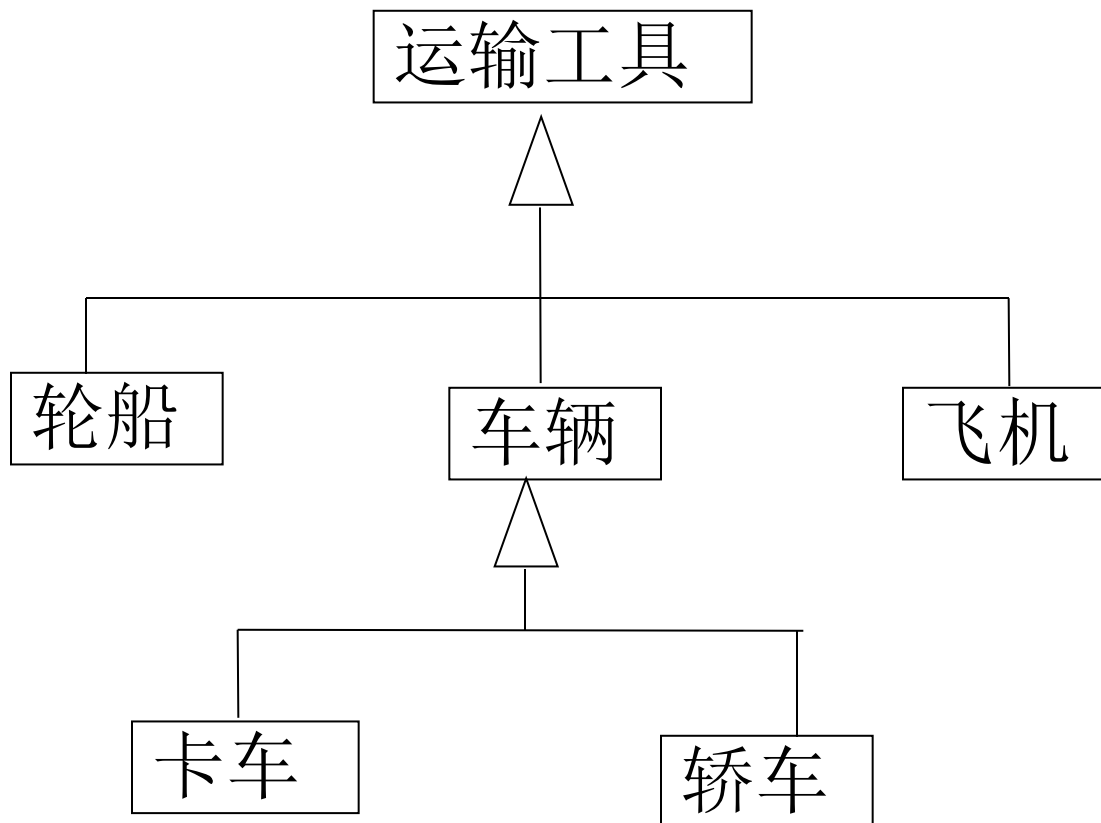
- 类可以有子类，子类继承父类的全部属性和操作，并允许添加自己的属性和操作
 - 继承是泛化和特殊的关系
 - 父类具有某事物的一般共性
 - 子类描述的事物在父类的基础上并比父类更特殊
- 继承可以有多样性，允许多于一个父类
- 继承允许多层，各层之间具有传递性；
- 继承的重要作用在于源代码的复用。



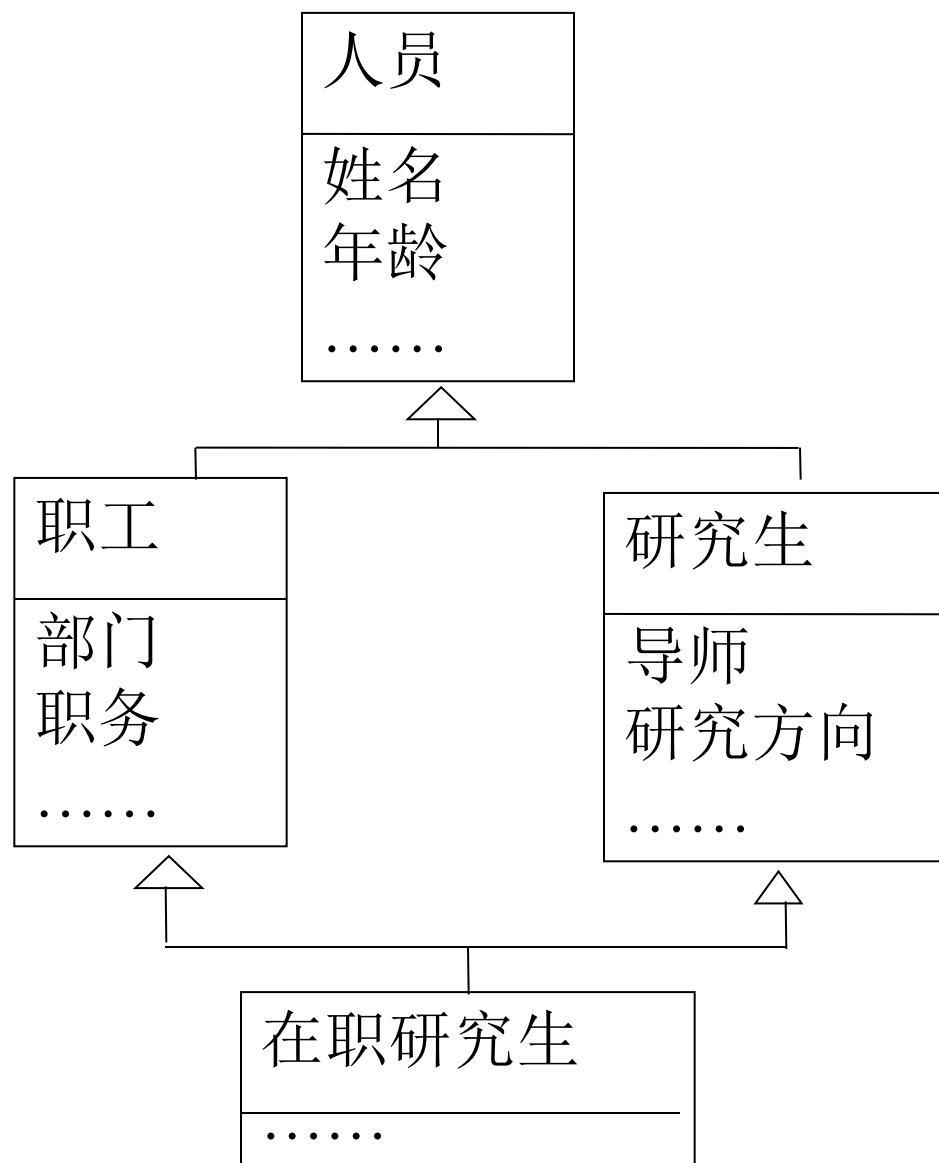
继承示例



多层继承示例1



多层继承示例2



(4) 多态性

- 在继承类结构中允许定义同名操作，同一个消息的响应可以执行不同的行为，即同一操作的多种形态
- 多态性更好地体现了操作语义的一致性，实现接口封装独立性和信息隐蔽的原则

多态的实现机制

■ 静态联编

- 编译时确定所访问对象的操作地址

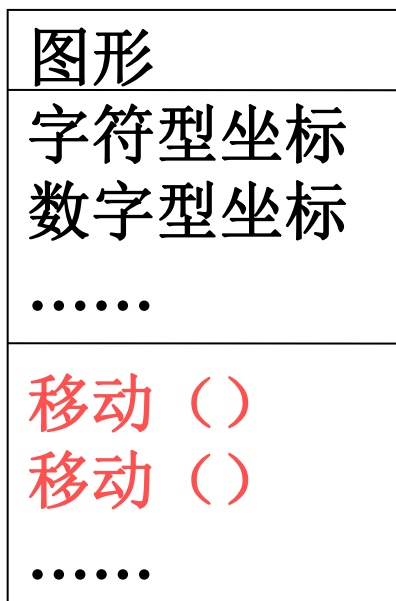
■ 动态联编（滞后联编 或 动态绑定）

- 编译时不确定所访问对象的操作地址，在运行时根据操作对象的不同再确定
- 动态联编使软件应用分布式发展有了重要突破，成为网络信息化、软件体系结构研究的主要支撑技术。

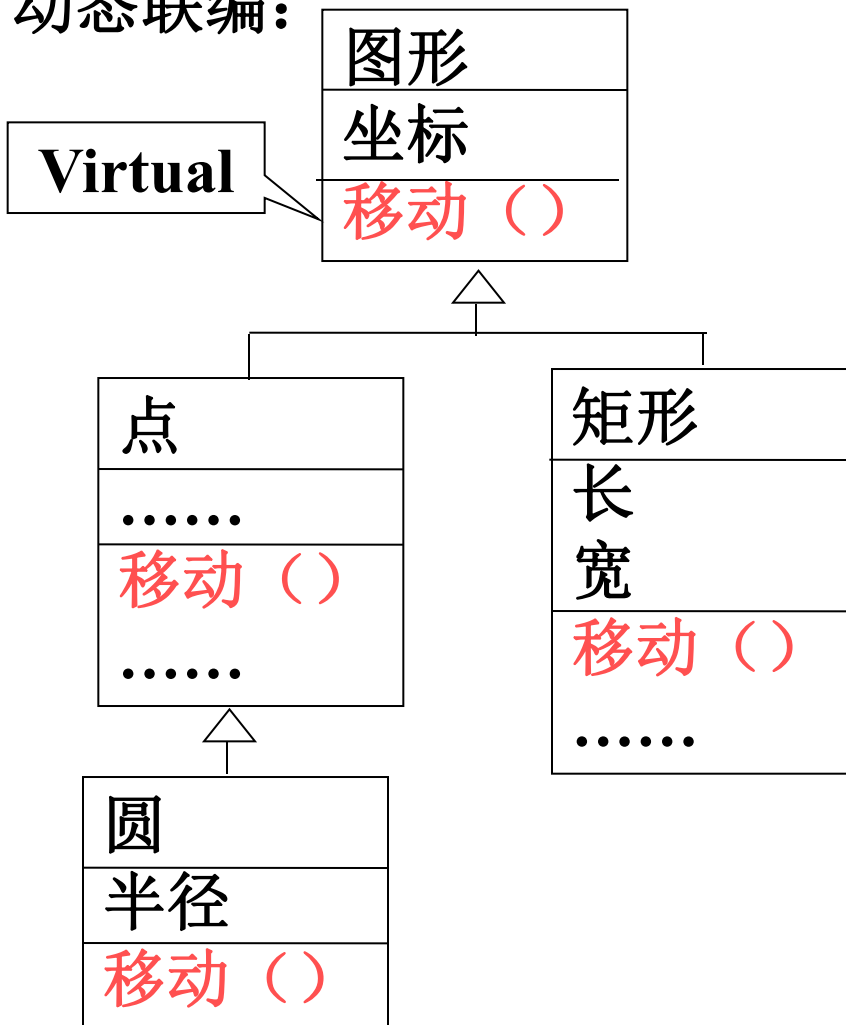


多态性示例

静态联编:



动态联编:



■ 可使用统一图形移动接口，采用移动 () 的动态联编



面向对象方法目标及实现机制

- 高可维护性
 - 对象类机制
- 可复用性
 - 继承机制
- 程序无关性
 - 多态性动态绑定机制



高可维护性

寻找可构造的**元素（类）**作为系统构造的基础，回避在不稳定基础上建造系统，使系统成为可构造和高可维。

- 1) 类是封装了操作的一个“代码级复用”程序模板，类的对象是系统的可构造元素
- 2) 采用消息机制执行对象的操作，回避了功能调用的过程性



可复用性

1) 对象语义一致性

- 功能的复用依赖于对功能的理解，对功能的描述是复杂和多义的，相比对象语义来说不容易复用。

2) 全方位复用

- 功能复用是代码级的，软件复用不但需要代码级的，还应该有序程序级的复用。
继承机制是源程序级的复用

程序/机器无关性

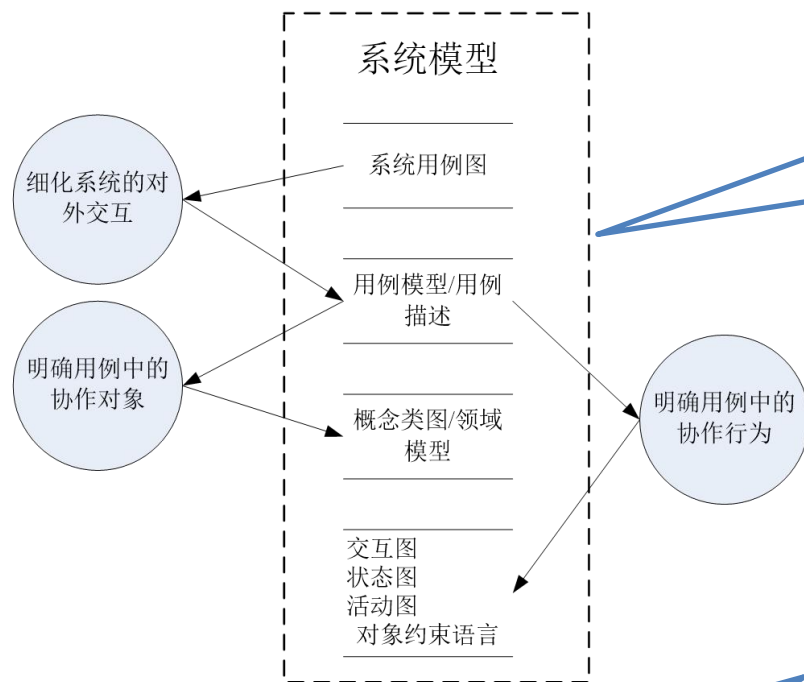
- 在任何机器环境下，使用任意程序语言，所编的程序都应该是逻辑通用的
 - 与机器相关的成分应该与处理逻辑无关。
- 类的多态性以及动态绑定技术
 - 提供了独立接口的实现技术
 - 这样可以将与机器相关的成分独立出来，为程序无关性奠定了基础。

面向对象分析与UML建模(1)

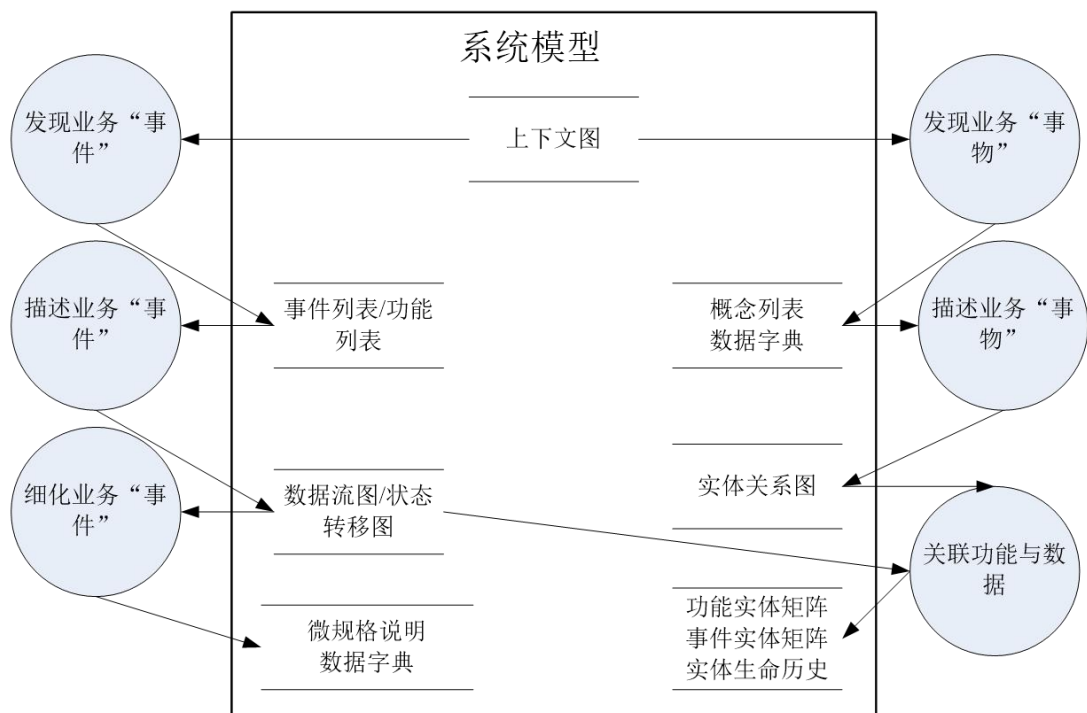
- 面向对象方法概述
- 面向对象分析 VS 结构化分析
- UML模型概述
- UML和Rose

面向对象分析 VS 结构化分析

继承、聚合、封装等,这使得软件开发者能更有效地思考问题,并以其他人也能看得懂的方式把自己的认识表达出来



从数据的角度对系统建模,主要通过过程建模(DFD图)和数据建模(ER图)描述数据流通系统时的整个流程和每一个步骤。



两种方法共同点

■ 抽象和分解

- 即对现有的现实问题进行不断地分解，同时也是对现实问题的一种抽象；
- 通过抽象降低软件模块之间的耦合性

■ 模块重用

- 在分解和抽象的基础上，进行模块重用

■ 半形式化的表示方法

- 精确描述需求
- 用户和程序员的共同理解

结构化分析方法的优缺点

■ 优点

- 建立的模型比较直观、简单、可读性好
- 利于沟通，有效提高软件质量

■ 缺点

- 数据为最关键的实体，表示为数据结构
- 数据是与程序（算法）分离
- 数据在系统的各层都可以被访问
 - 单个数据的修改会导致整个系统程序的变化
 - 程序的耦合性的增加

面向对象分析方法的优缺点

■ 优点：高可重用/重构/变更性

■ 封装：数据不再贯穿整个程序，而是成为各个模块的私有属性

- 数据不和算法分离 ----- 更符合现实情况
- 大大降低了系统模块间的耦合性

■ 继承和多态

- 体现了现实世界的多样性

■ 缺点

- 一个类的内部更为复杂&UML模型更为复杂
- 需要掌握更复杂的建模知识

面向对象分析与UML建模(1)

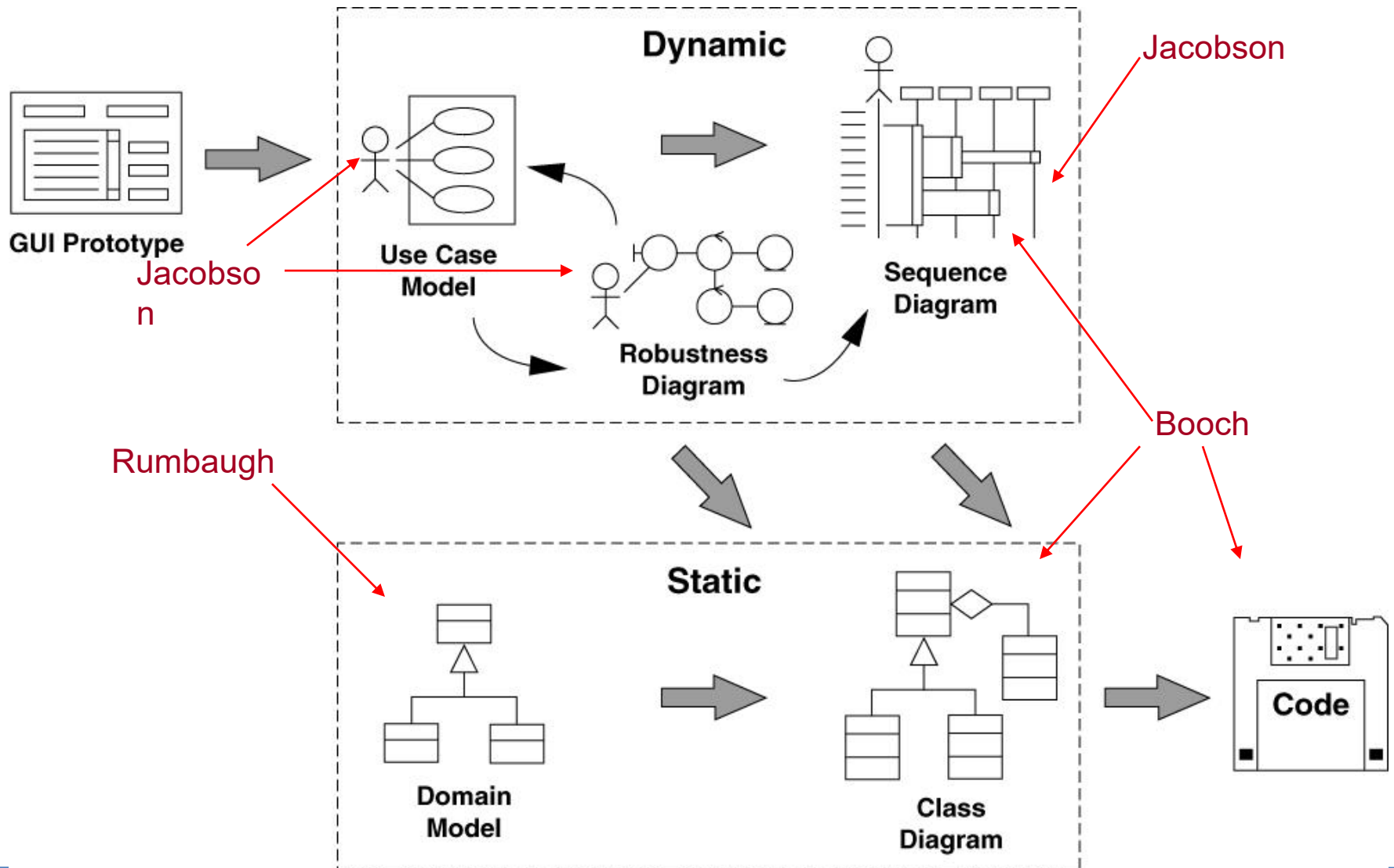
- 面向对象方法概述
- 面向对象分析 VS 结构化分析
- UML模型概述
- UML和Rose

面向对象建模

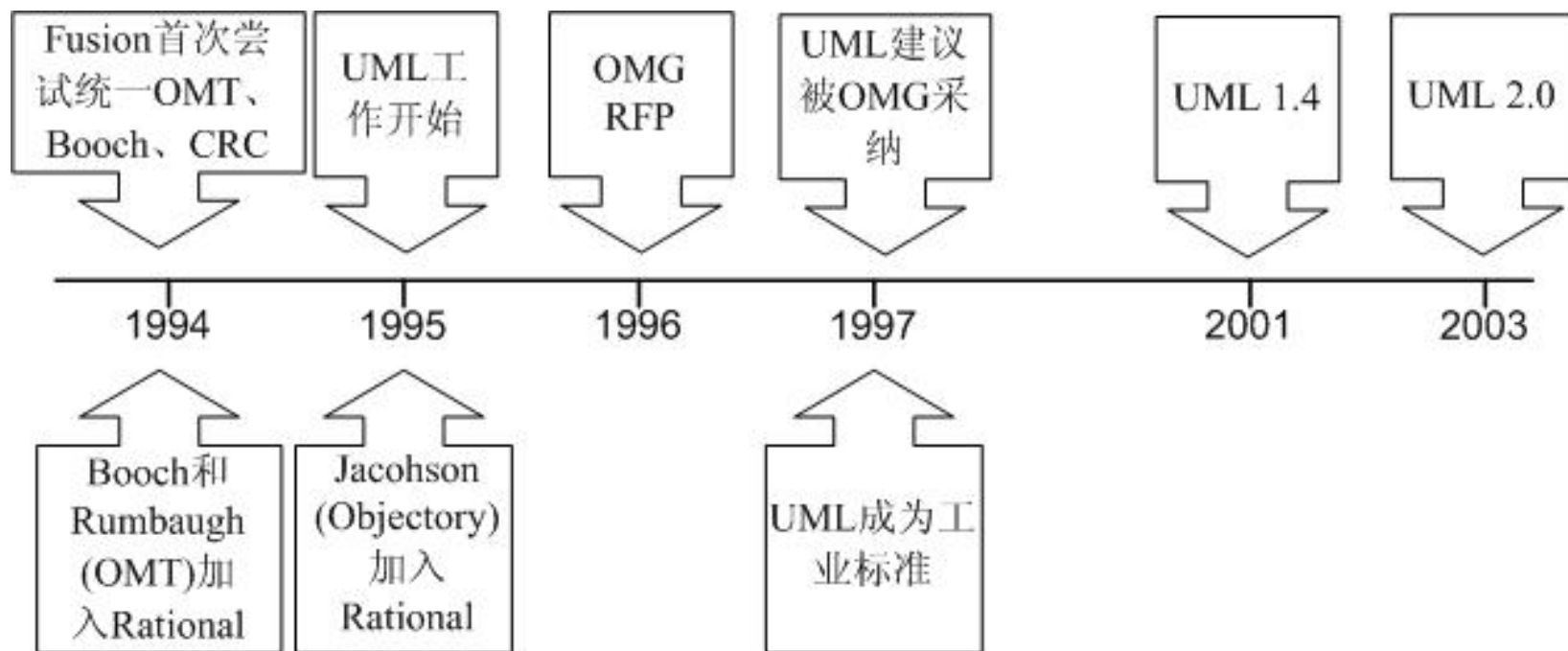
- UML是一种统一的、标准化的建模语言
- UML是一种应用面很广泛的建模语言
- 面向对象建模：一种用于辨识系统环境中的对象及这些对象之间关系的技术
 - OMT (James Rumbaugh)
 - Booch方法(Grady Booch)
 - OOSE (Ivar Jacobson)

} UML

面向对象分析—UML



UML发展历程



UML的特性与发展现状

- UML是一种Language（语言）
- UML是一种Modeling（建模）Language
- UML是Unified（统一）Modeling Language
- 已进入全面应用阶段的事实标准
- 应用领域正在逐渐扩展，包括嵌入式系统建模、业务建模、流程建模等多个领域
- 成为“产生式编程”的重要支持技术：MDA、可执行UML等

软件需求工程中的UML

■ 需求获取

- 对捕获的需求进行整理和分析的工具，辅助开发人员和用户进行沟通

■ 需求分析

- 包含高层设计（架构模型）和详细设计模型，用于统一开发人员、沟通设计信息

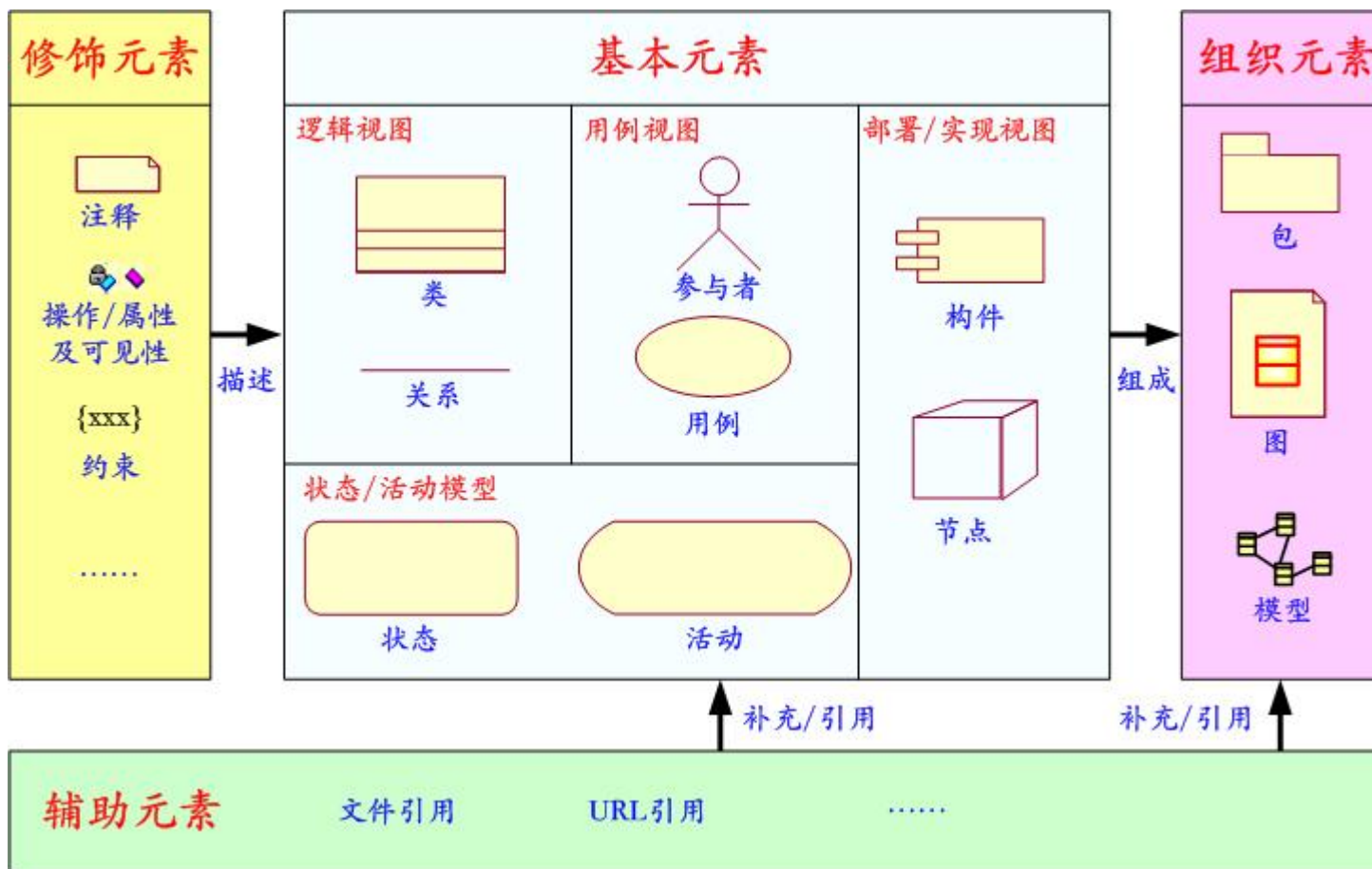


面向对象分析（1）

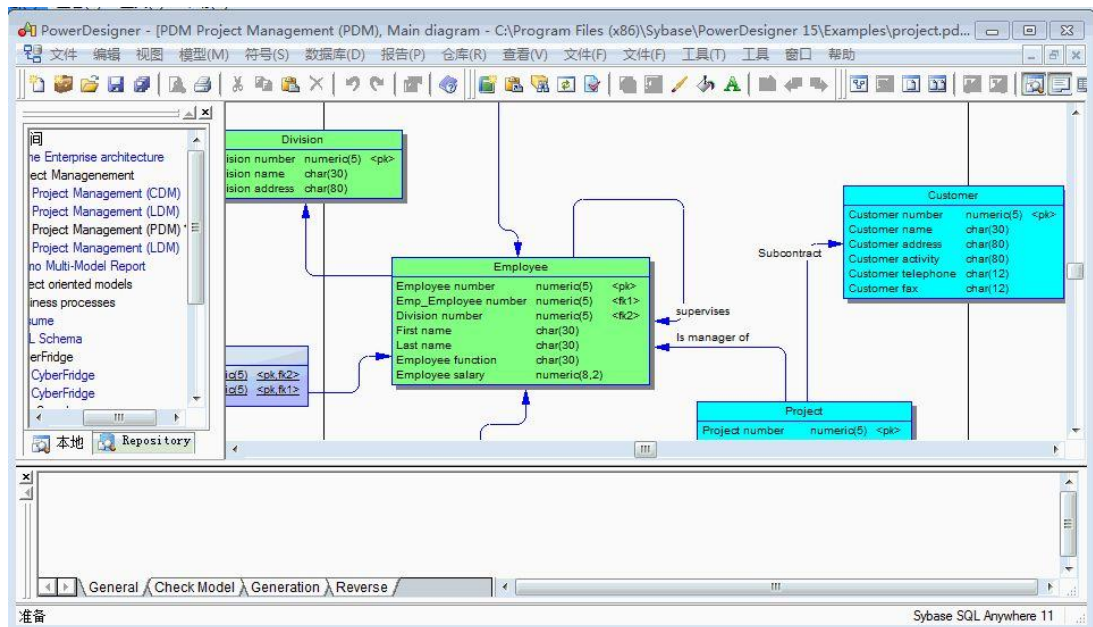
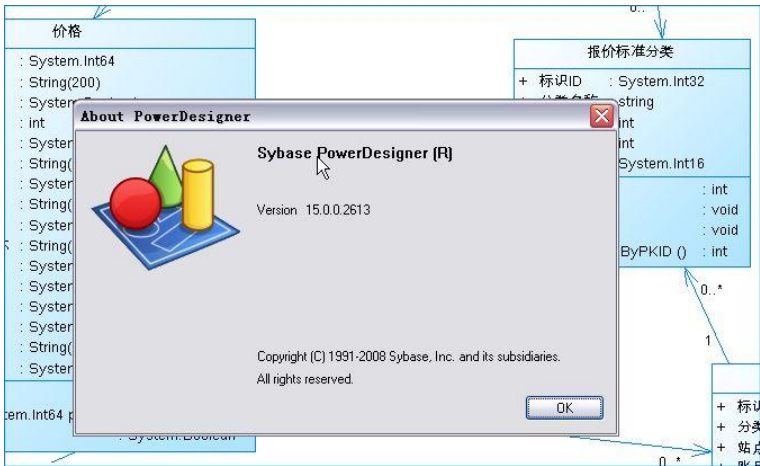
- 面向对象方法概述
- 面向对象分析 VS 结构化分析
- UML模型概述
- UML和Rose

Rational Rose UML建模工具

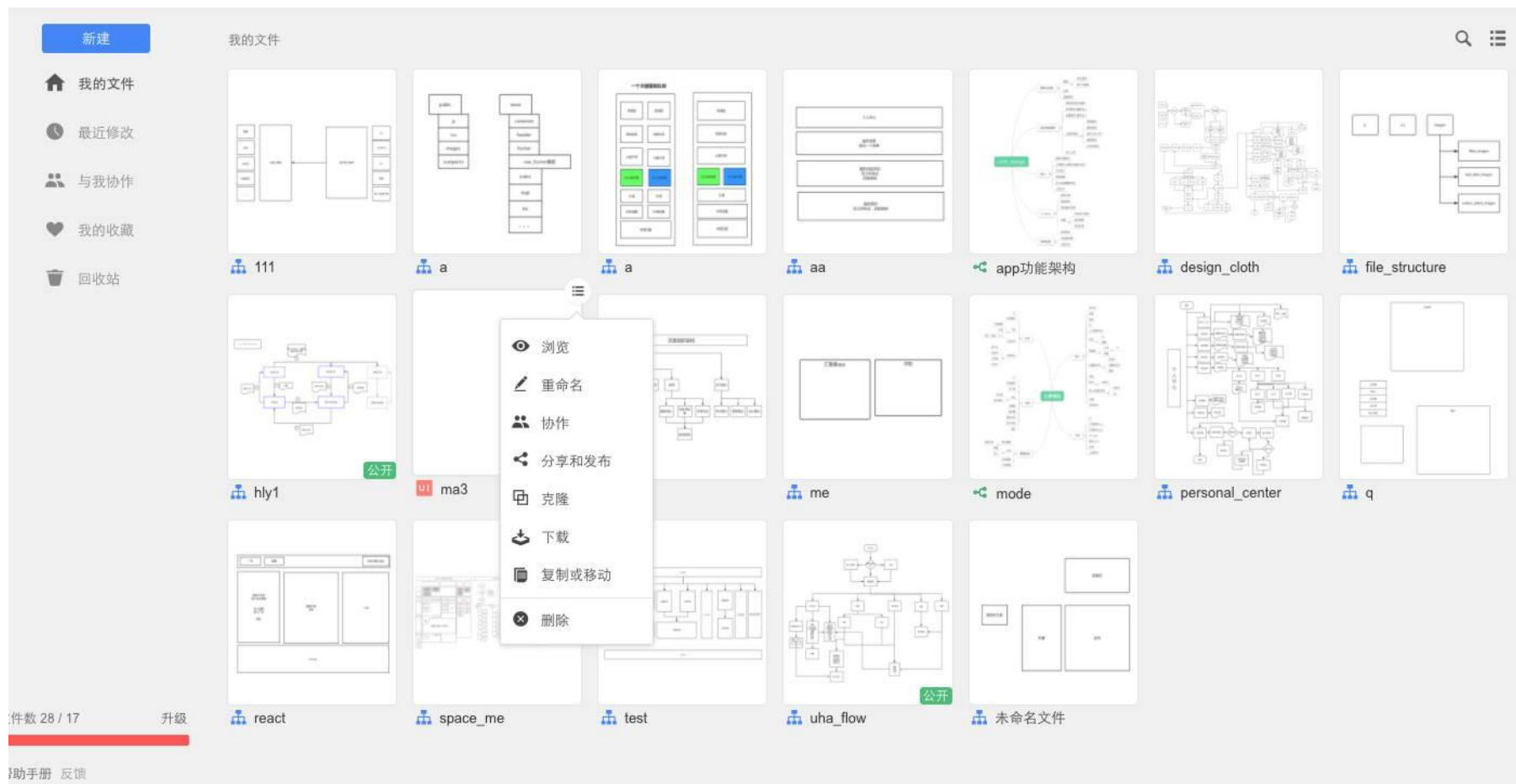
■ Rose下的UML基本元素



其它UML建模工具: PowerDesigner



其它UML建模工具: ProcessOn



Rose模型组织元素

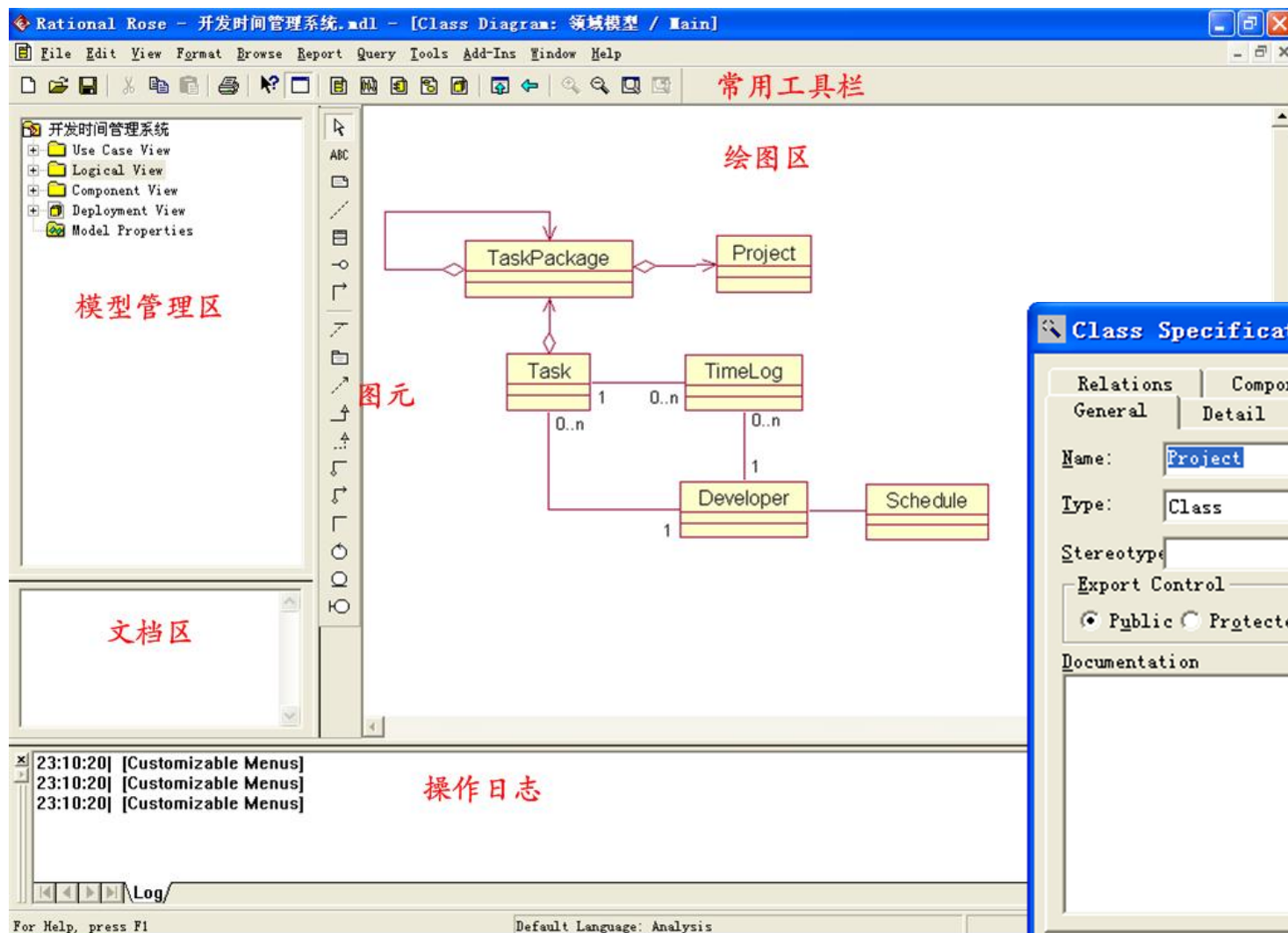
- 图：最直接的一种组织元素
- 模型：
 - 业务分析模型 --系统分析模型
 - 设计模型 --实现模型
 - 测试模型
- 视图
 - 逻辑视图 --进程视图
 - 实现视图 --部署视图
- 包：视图、模型、子系统、层、包

视图、模型与图

视图	模型	图
用例视图	业务/系统分析模型(为主)	用例图、交互图、状态机图、类图、交互概述图
	设计模型	交互图、状态机图、类图、交互概述图
逻辑视图	业务分析模型	类图
	设计分析模型(为主)	类图、交互图、状态机图、对象图、包图
	设计模型	活动图、交互图、活动图、交互概述图、状态机图、对象图
实现视图	设计模型	构件图、部署图
部署视图	设计模型	部署图



Rose操作基础



Class Specification for Project

Relations Components Nested Files

General Detail Operations Attributes

Name: Project Parent: 领域模型

Type: Class

Stereotype

Export Control

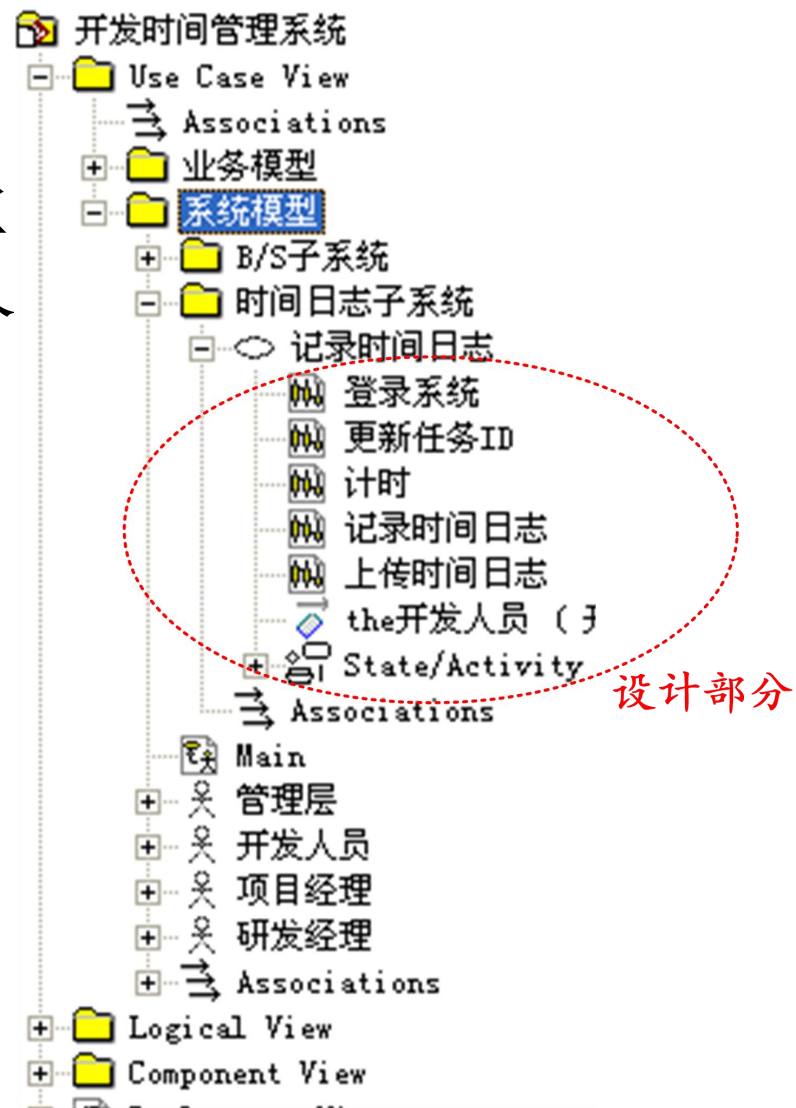
☒ Public ☐ Protected ☐ Private ☐ Implementation

Documentation

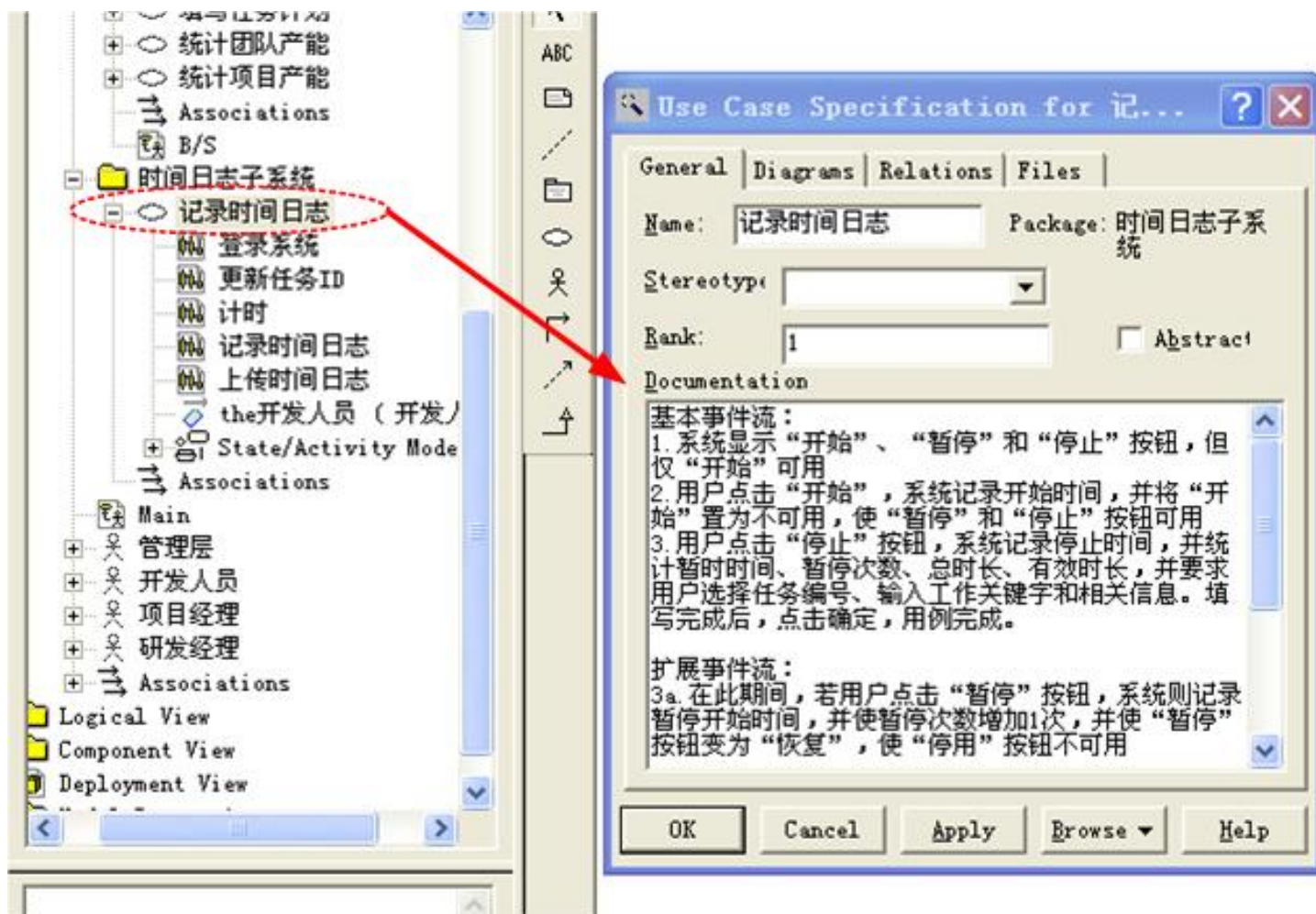
OK Cancel Apply Browse Help

UML用例图组织方法

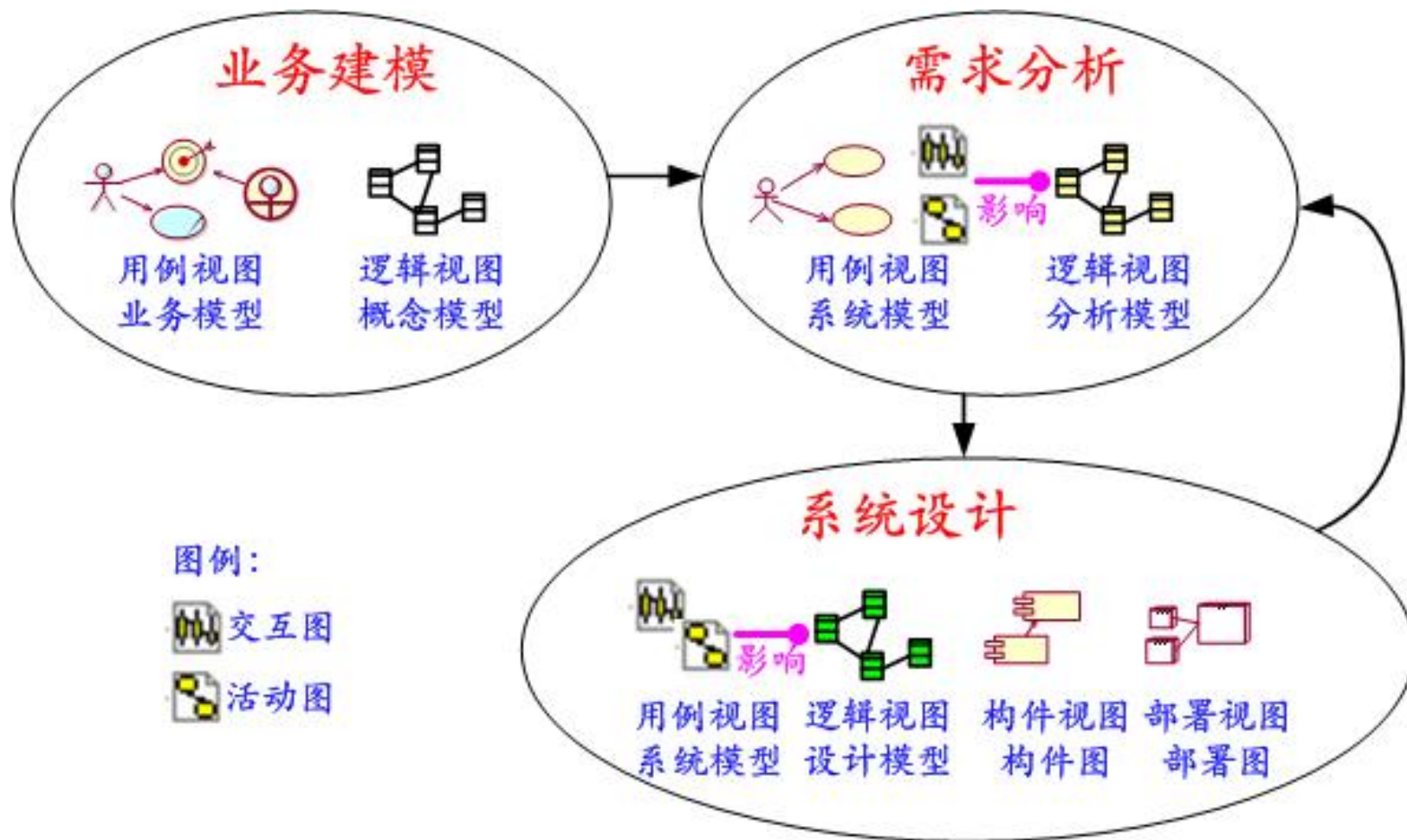
- 在Use Case View中包含了“业务模型”和“系统模型”两个包，这分别是指
 - 业务建模工作流产生的业务分析模型
 - 需求及分析工作流产生的系统分析模型



用例图的细化



Rose模型创建流程



模型发布

Rose Web Publisher - Microsoft Internet Explorer

文件(F) 编辑(E) 查看(V) 收藏(A) 工具(T) 帮助(H)

地址(D) D:\test\index.htm

Display documentation

- Use Case View
 - 业务模型
 - 系统模型
 - B/S子系统
 - 时间日志子系统
 - 记录时间日志
- Logical View
 - 领域模型
 - 设计模型
 - B/S子系统
 - 时间日志子系统
 - UI
 - BusinessRule
 - DataAccess
 - Package关系
 - Main
 - Associations
- 分析模型
 - Main
 - Associations
- Component View
- Deployment View
- Project Properties

Home

Use Case 记录时间日志

Documentation

基本事件流:

1. 系统显示“开始”、“暂停”和“停止”按钮，但仅“开始”可用
2. 用户点击“开始”，系统记录开始时间，并将“开始”置为不可用，使“暂停”和“停止”按钮可用
3. 用户点击“停止”按钮，系统记录停止时间，并统计暂时时间、暂停次数、总时长、有效时长，并要求用户选择任务编号、输入工作关键字和相关信息。填写完成后，点击确定，用例完成。

扩展事件流:

3a. 在此期间，若用户点击“暂停”按钮，系统则记录暂停开始时间，并使暂停次数增加1次，并使“暂停”按钮变为“恢复”，使“停用”按钮不可用

3a1. 当用户点击“恢复”按钮，用当前时间减去暂停开始时间得到本次暂停时间，并累加到“暂停时间”时间中，并使“恢复”按钮变为“暂停”，使“停用”按钮恢复可用

约束:

时间记录程序应以离线式工作，该程序会自动连接服务器，完成时间日志上传的工作，如果未能连接服务器，则在本机暂存时间日志

Parent Package	时间日志子系统	Abstract	No
Rank			

Associations --Not Named--

State/Activity Models [State/Activity Model](#)

小应用程序 IncrementalTOC started

Rose Web Publisher

Selector

- ☒ Use Case View
- ☒ Logical View
- ☒ Component View
- ☒ Deployment View

Level of Detail

- ☐ Documentation Onl
- ☐ Intermediat
- ☒ Full

Notation

- ☐ Booch
- ☐ OMT
- ☒ UML

Include Inherited I

Include Propert:

Include Associations in Br

Include Document Wrapping in Browser

HTML Root File Name

Elapsed Time: 00:00:00

Publish

Diagrams...

Help

Close

Preview

本节小结

- 面向对象分析是90年代之后的主流分析方法，它以UML为基础，综合使用了多种不同的分析技术
- 面向对象分析 VS 结构化分析
- UML模型概述
- Rose建模工具