



需求分析与原型化

清华大学软件学院 刘强





1

软件分析建模

2

统一建模语言 UML

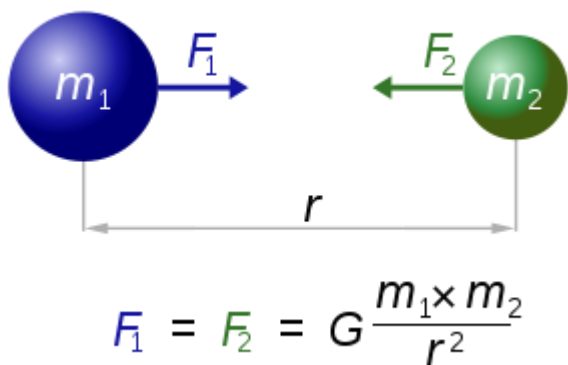
3

需求原型与文档化

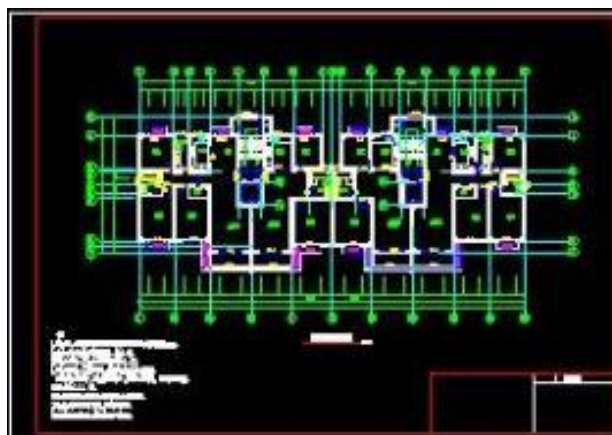
模型的概念

模型是对系统或事物的简化、类比和抽象，将其主要特征用形式化的语言或图形进行描述，用于研究和揭示系统或事物的形态和本质。

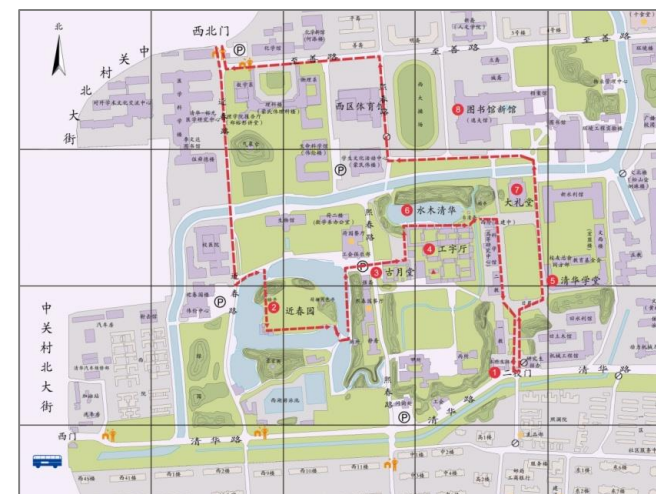
万有引力定律



建筑设计



清华大学

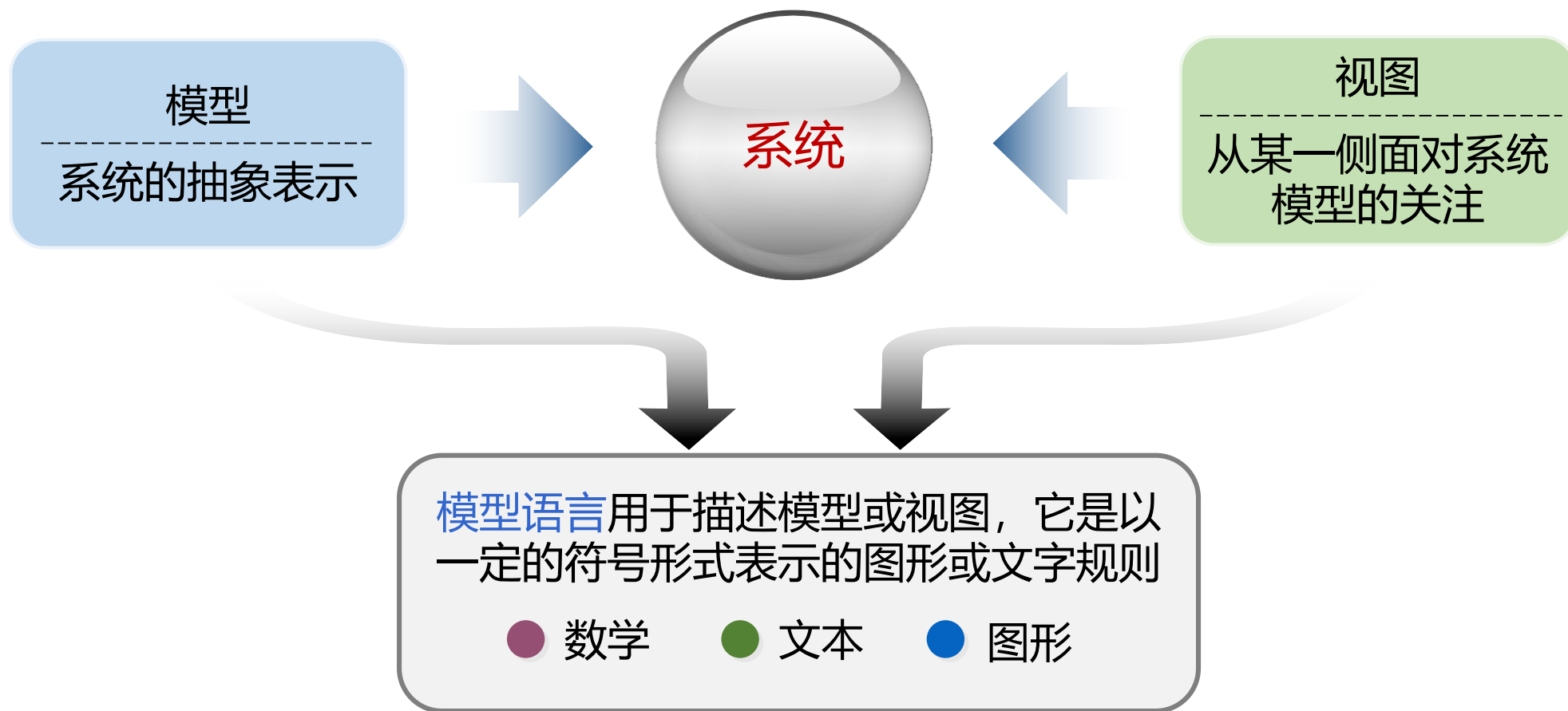


模型的重要性



- 模型是对现实的简化，可以帮助人们按照实际情况对系统进行可视化。
- 模型可以更早地发现问题或遗漏的地方，为代码生成提供依据。
- 一个好的模型包括重要的特征或因素，而忽略不相干的细节。
- 每一个系统可以从不同的方面使用不同的模型进行描述，因此每个模型都是对系统从语义上近似的抽象。
- 模型可以详细说明系统的结构和行为，结构模型侧重于系统的组织，行为模型侧重于系统的动作。

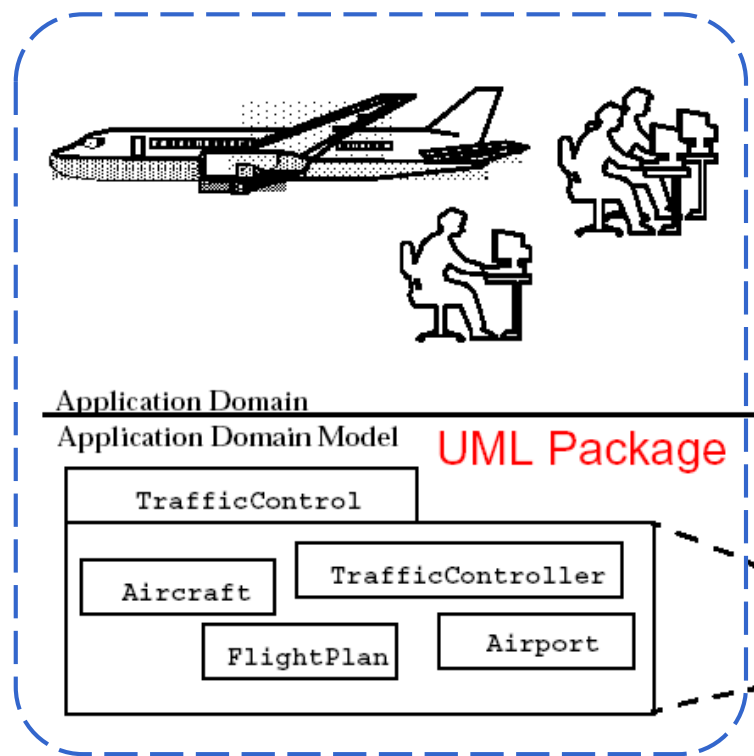
系统、模型和视图



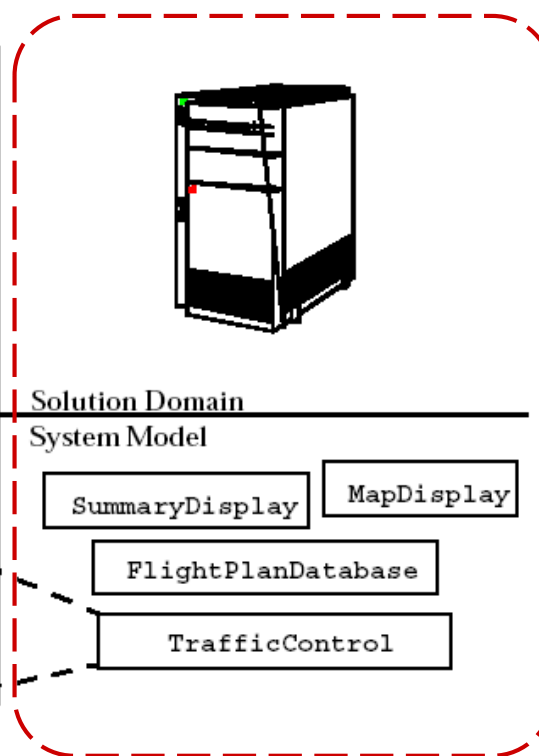
软件分析建模

软件分析建模从不同侧面描述软件系统的数据信息、处理功能以及运行的外部行为，可以有效地简化和处理复杂性，将需求映射到软件结构中。

问题域模型



求解域模型

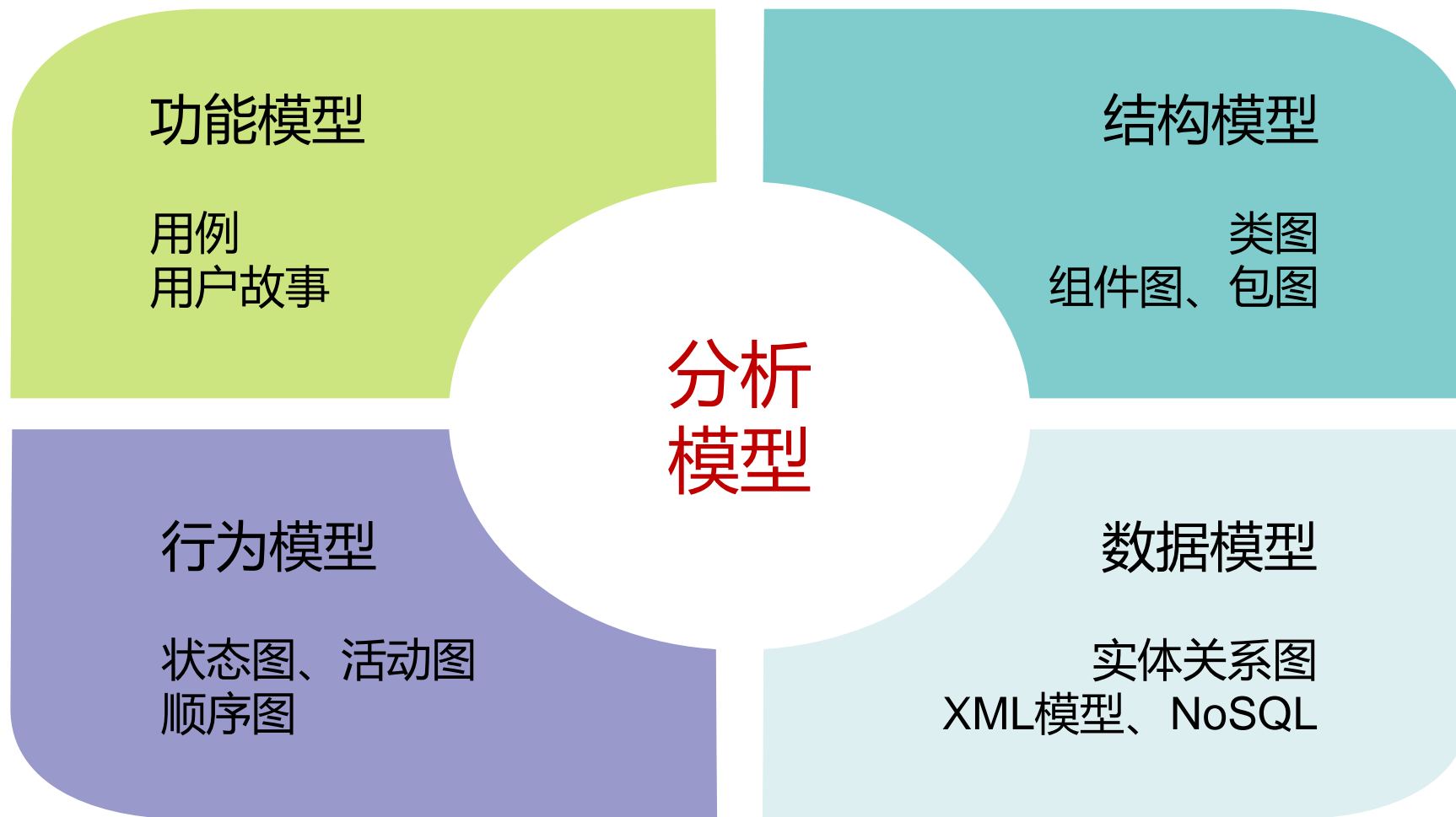


软件分析建模

软件分析建模从不同侧面描述软件系统的数据信息、处理功能以及运行的外部行为，可以有效地简化和处理复杂性，将需求映射到软件结构中。

一张图胜过千言万语！

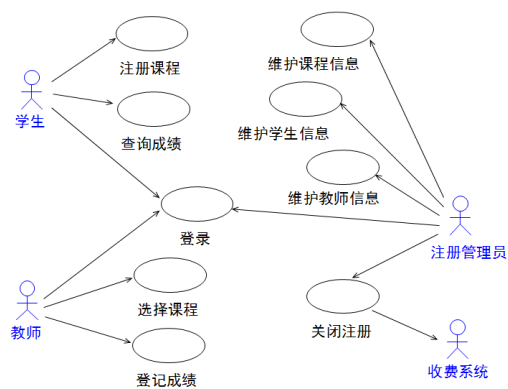
- UML常用图包括用例图、类图、状态图、顺序图等
- 流程图可以用于表示算法的思路，也可以表示事务处理的加工过程
- ER图（实体-关系图）用于建立数据模型，它是关系数据库设计的一种工具



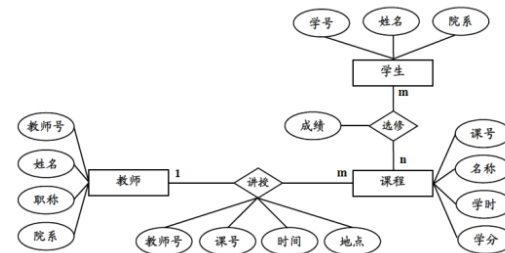
软件分析建模



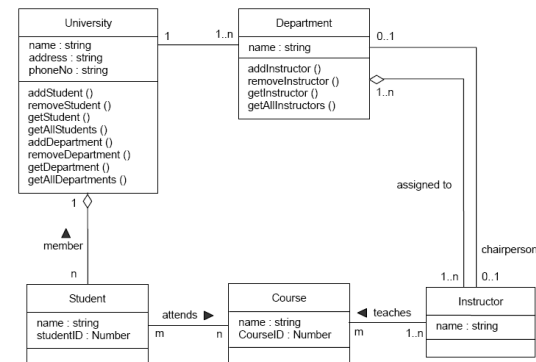
功能模型（用户故事）



功能模型（用例图）



数据模型（ER图）



结构模型（类图）

1

软件分析建模

2

统一建模语言 UML

3

需求原型与文档化

统一建模语言 UML

统一建模语言 (Unified Modeling Language, 简称UML) 是由面向对象领域的三位著名方法学家Grady Booch、James Rumbaugh和Ivar Jacobson共同提出的。



Booch

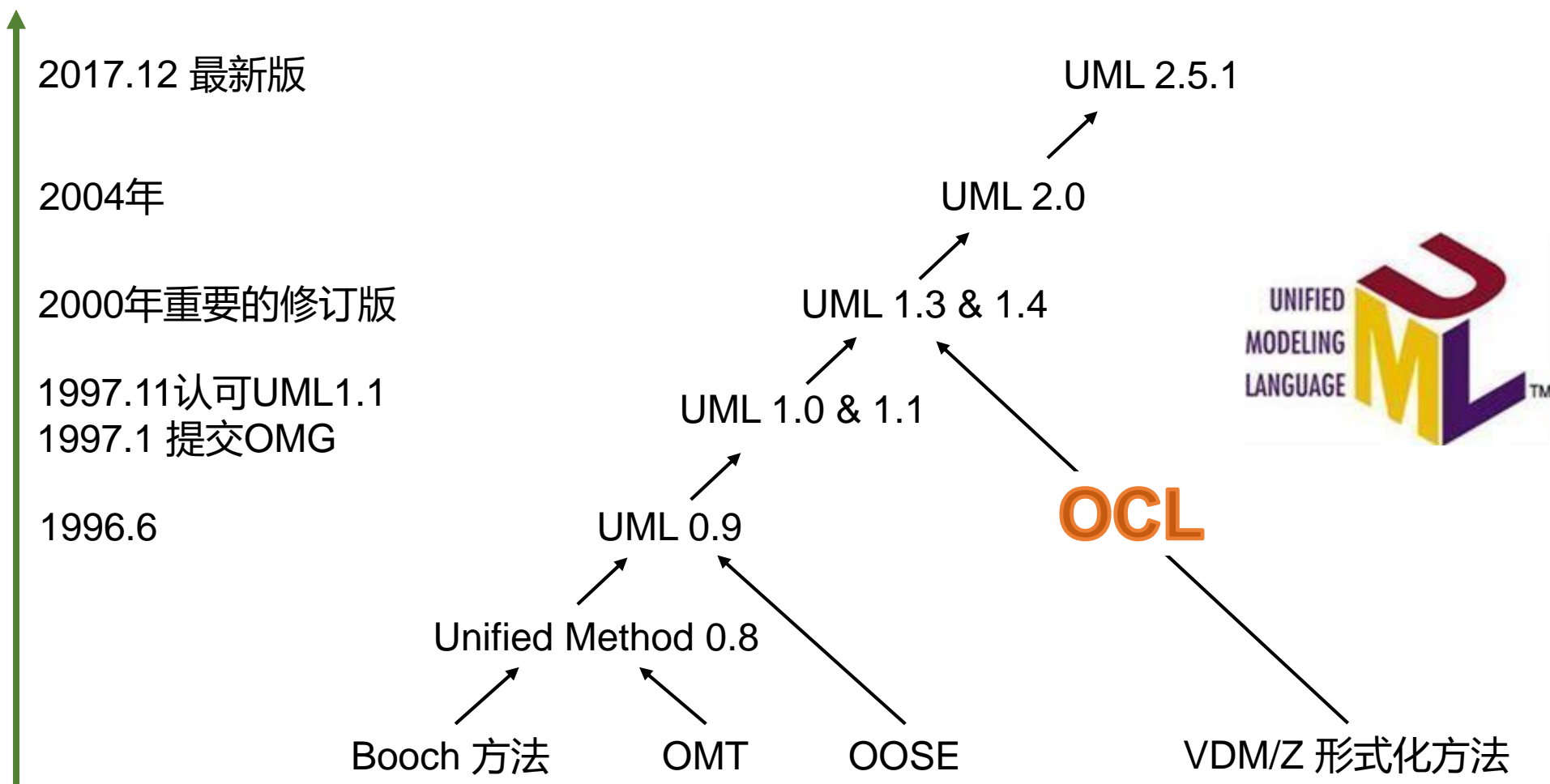


Rumbaugh

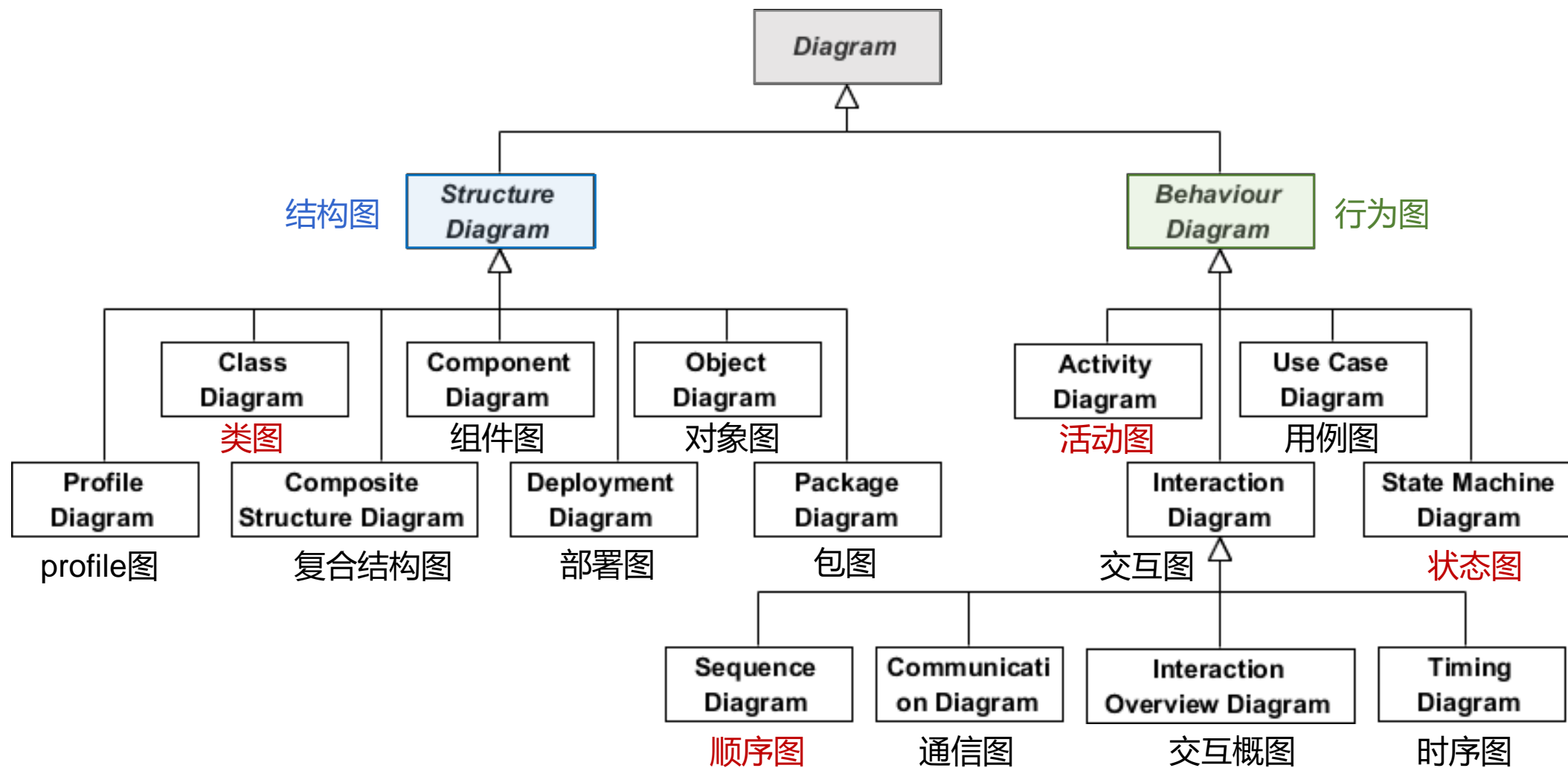


Jacobson

UML 发展历史

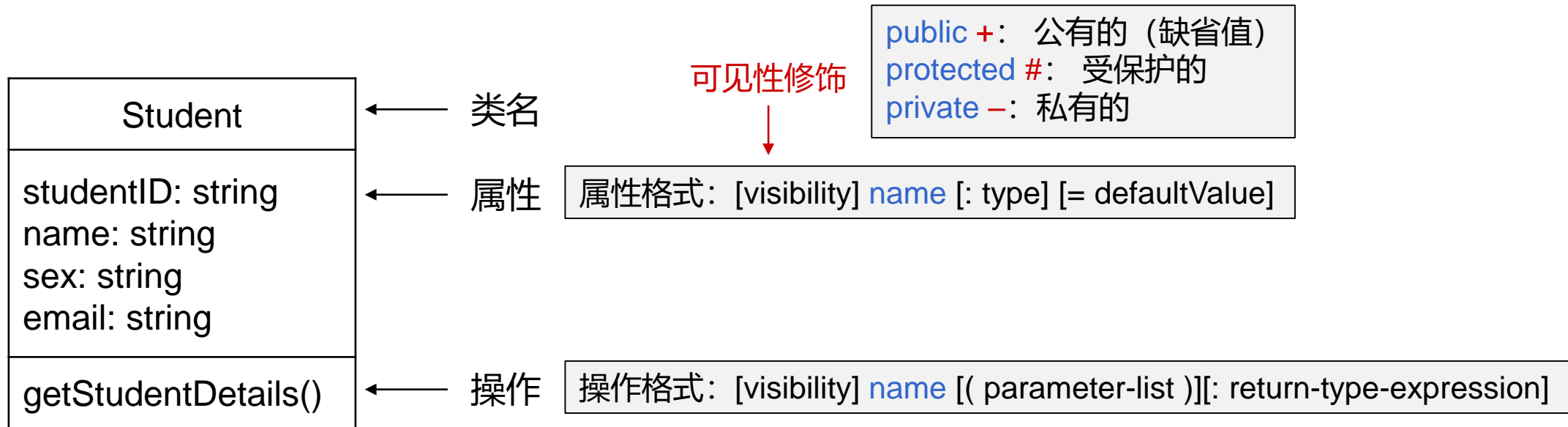


UML 模型



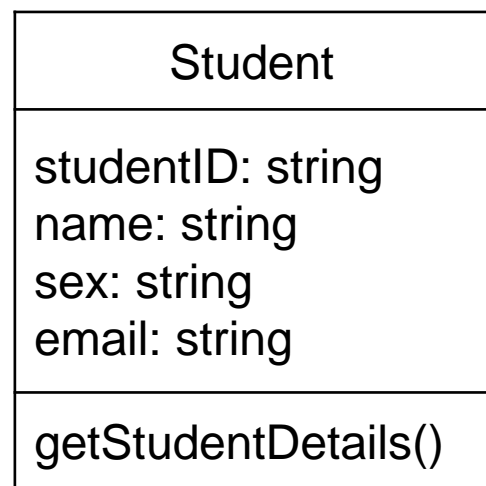
类图

类图 (Class Diagram) 描述系统的静态结构，表示系统中的类、类与类之间的关系以及类的属性和操作。



类图

类图 (Class Diagram) 描述系统的静态结构，表示系统中的类、类与类之间的关系以及类的属性和操作。



← 类名

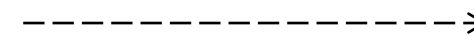
← 属性

← 操作

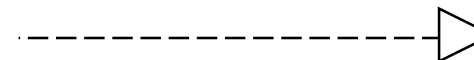
关联关系 (Association)



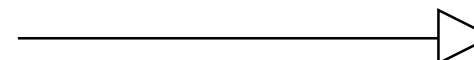
依赖关系 (Dependency)



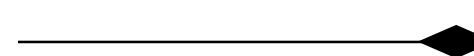
实现关系 (Implementation)



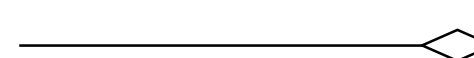
泛化关系 (Generalization)



组合关系 (Composition)

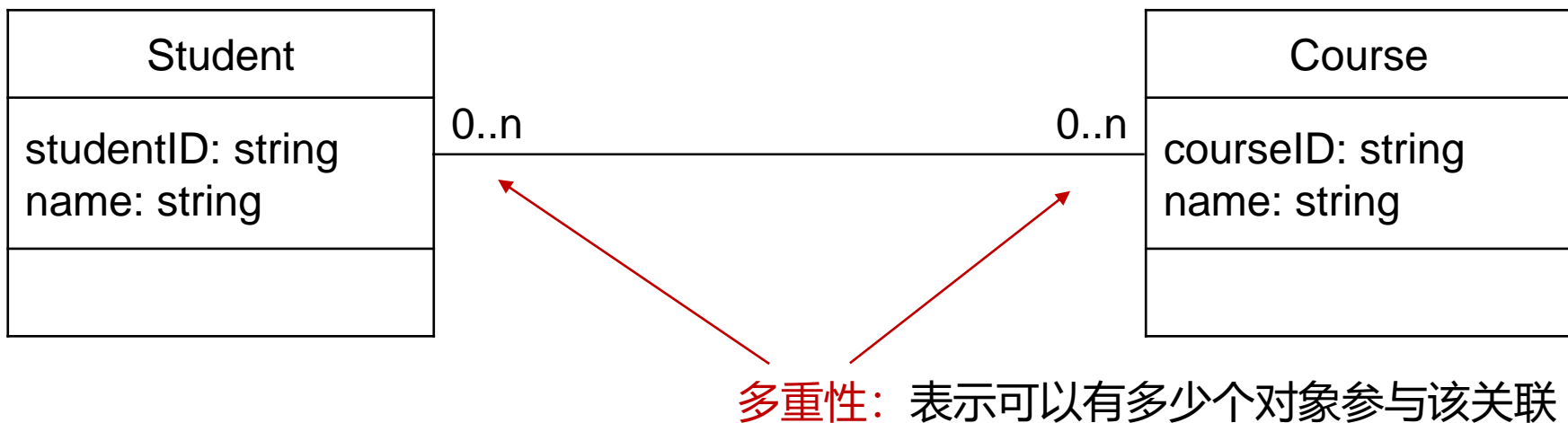


聚合关系 (Aggregation)



UML：关联

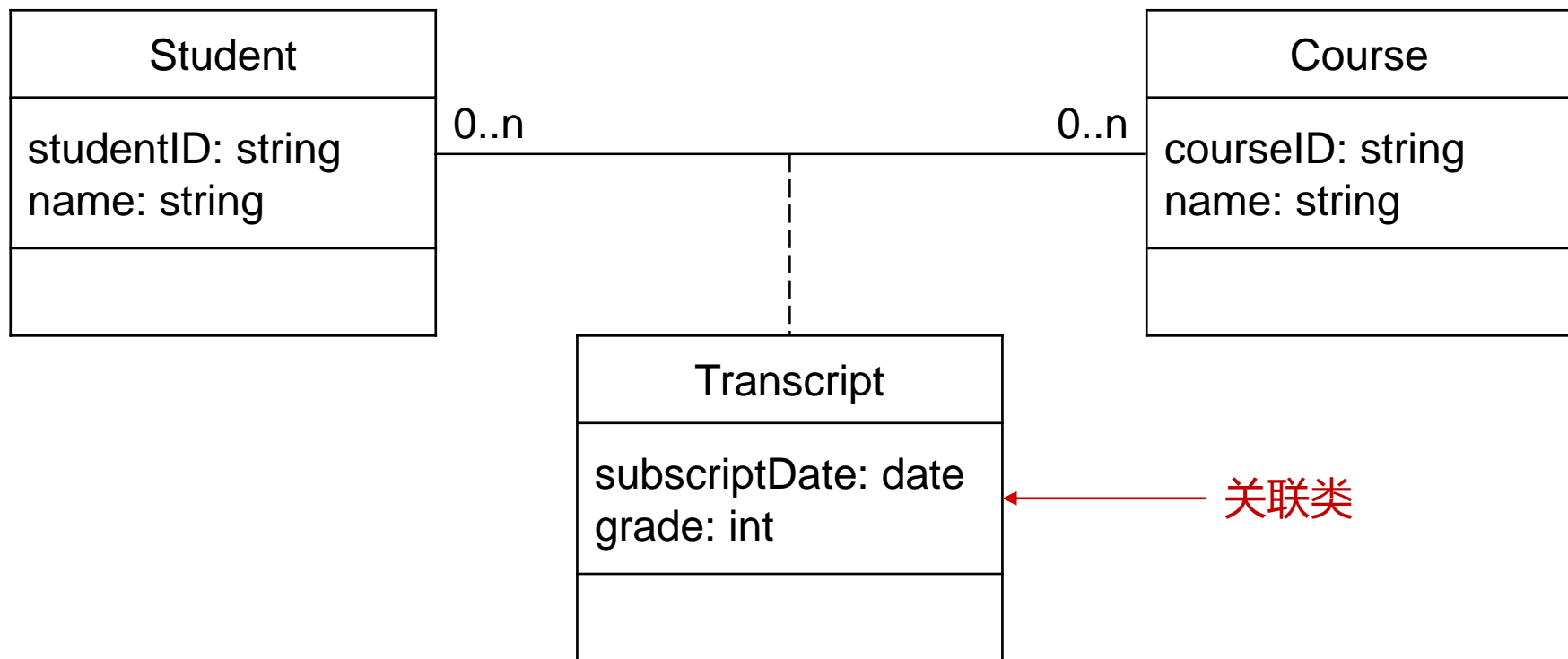
关联 (Association) 是一种结构化关系，用于表示一类对象与另一类对象之间的联系。



- 固定值：3
- 许多值：n 或者 *
- 区间：0..1 或者 3..n
- 集合：2,4,8

UML：关联类

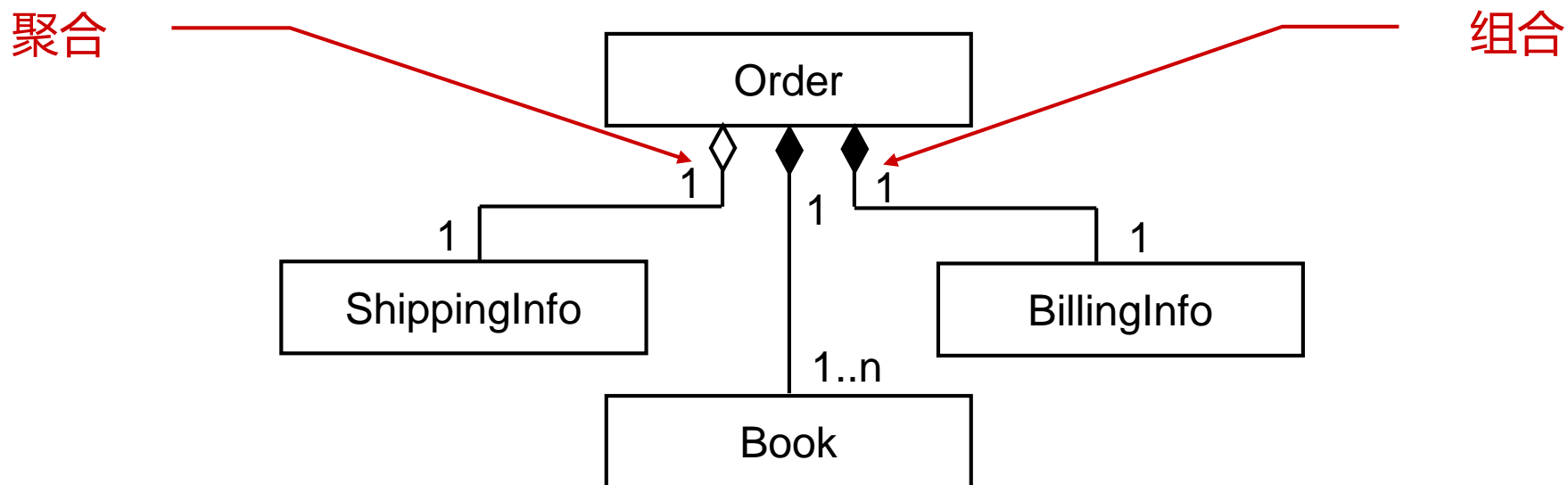
关联类是一种具有关联特性和类特性的建模元素。当两个类具有多对多关系时，有时存在一些属性，它们不能简单地放入任何一个类中。



UML：聚合和组合

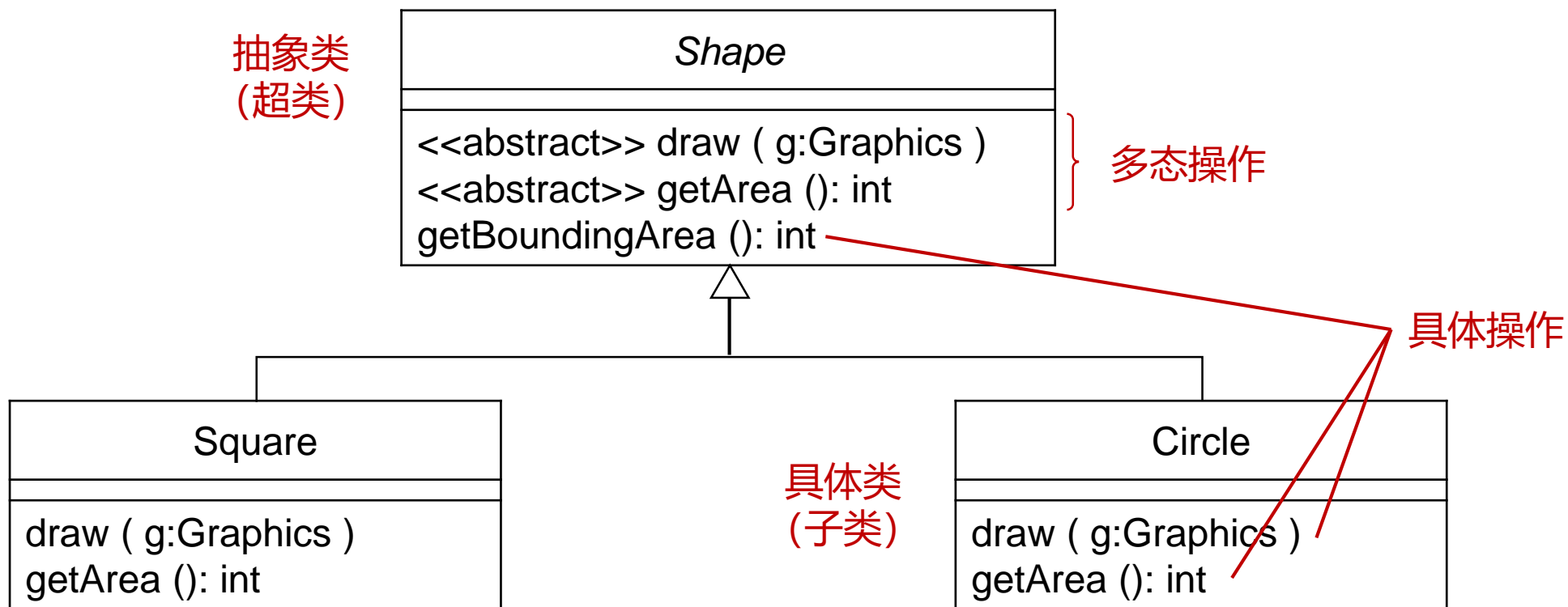
聚合 (Aggregation) 是一种特殊形式的关联，它表示类之间的整体与部分的关系。

组合 (Composition) 是一种特殊形式的聚集，其中的整体与部分具有同样的生存期。



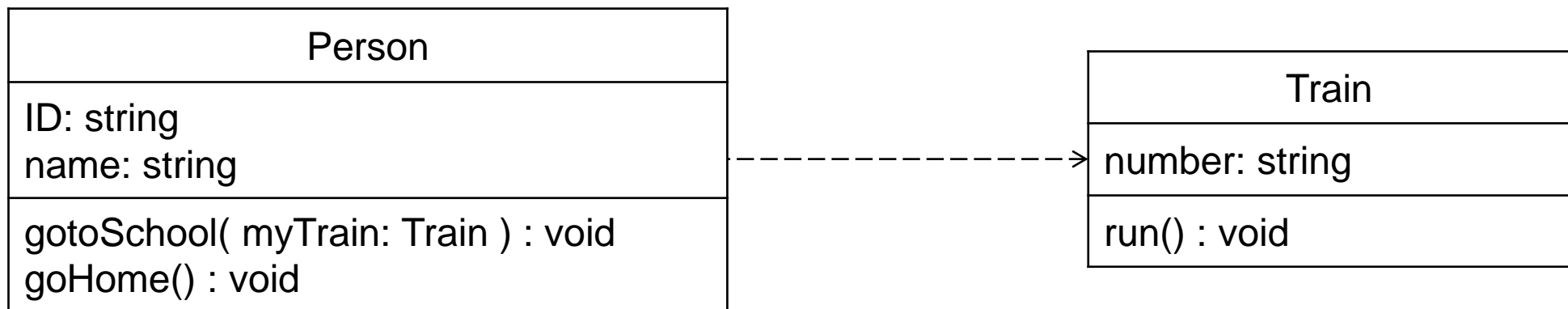
UML: 泛化

泛化 (Generalization) 是一种特殊/一般的关系，简单地说就是类之间的继承关系。



UML：依赖

依赖（Dependency）是一种使用关系，即一个类A的实现需要另一个类B的协助。这种关系具有偶然性和临时性，但是类B的变化会影响到类A。



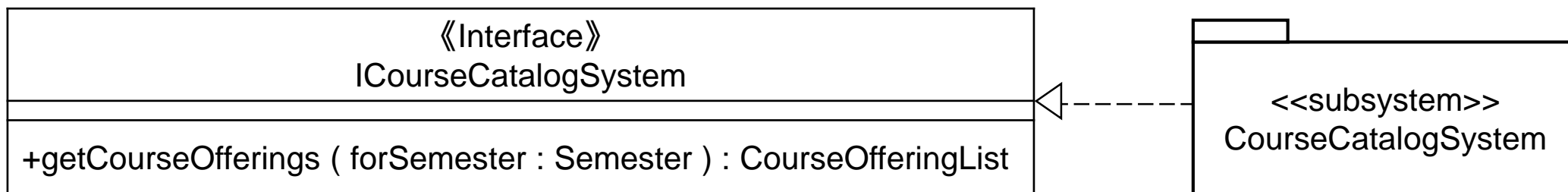
在代码层面有三种表现形式

- 将一个类的对象作为另一个类中方法的参数
- 在一个类的方法中将另一个类的对象作为其局部变量
- 在一个类的方法中调用另一个类的静态方法

UML：实现

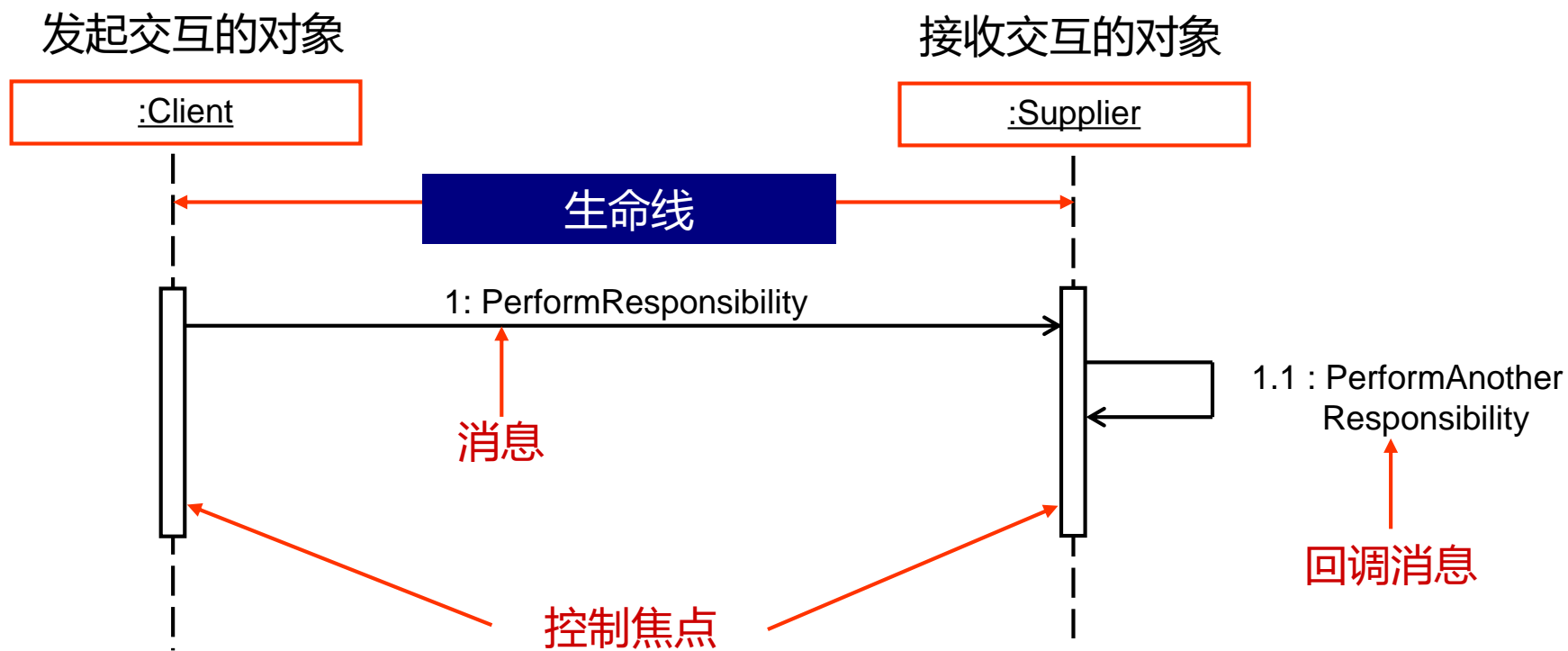
实现 (Realization) 指的是一个类实现接口（可以是多个）的功能，它是类与接口之间最常见的关系。

接口 (Interface) 是一种特殊的类，具有类的结构但不可被实例化，只可以被实现。接口表示的是一系列的操作集合，它指定了一个类所提供的服务。

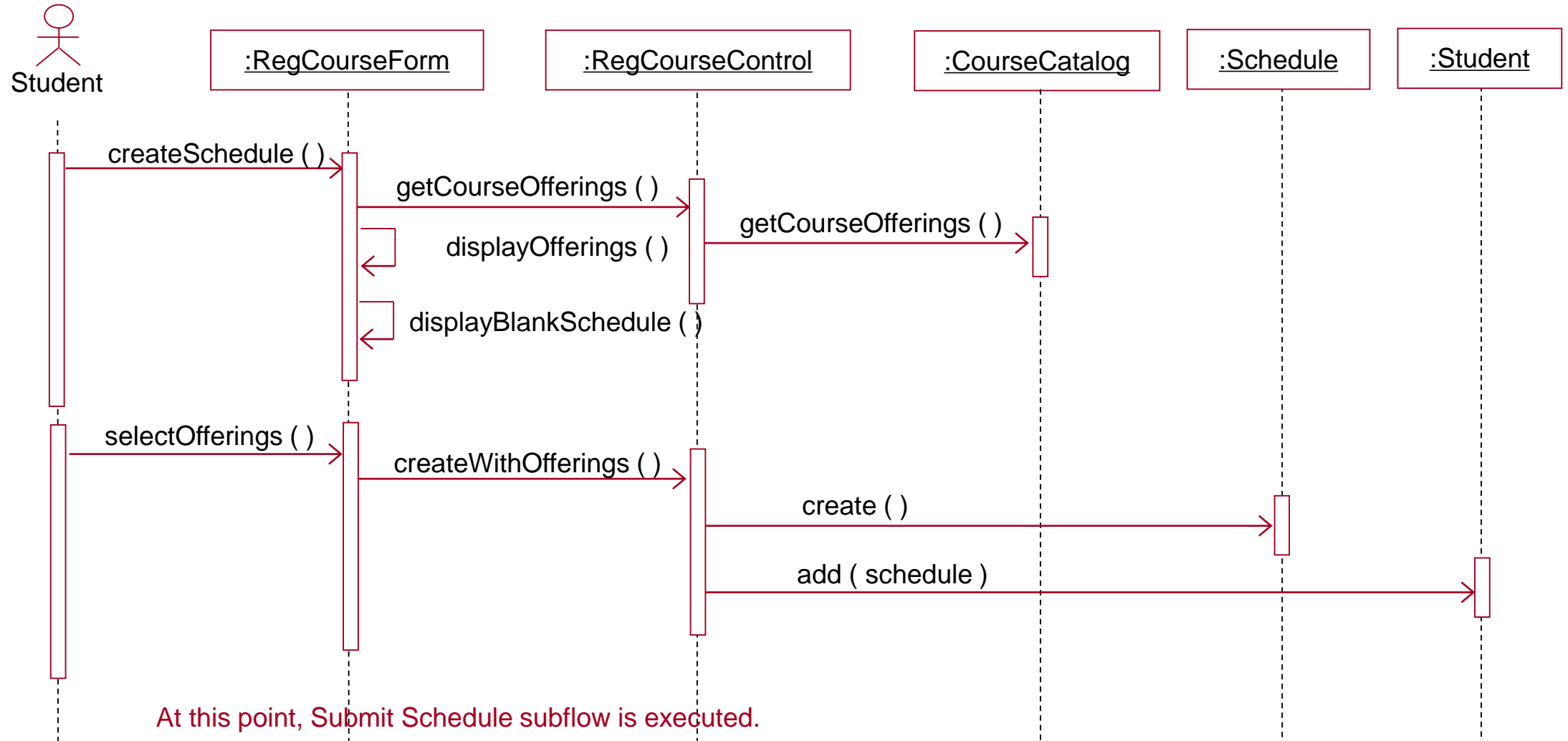


UML 顺序图

顺序图 (Sequence Diagram) 表示完成某项行为的对象之间传递消息的时间顺序。

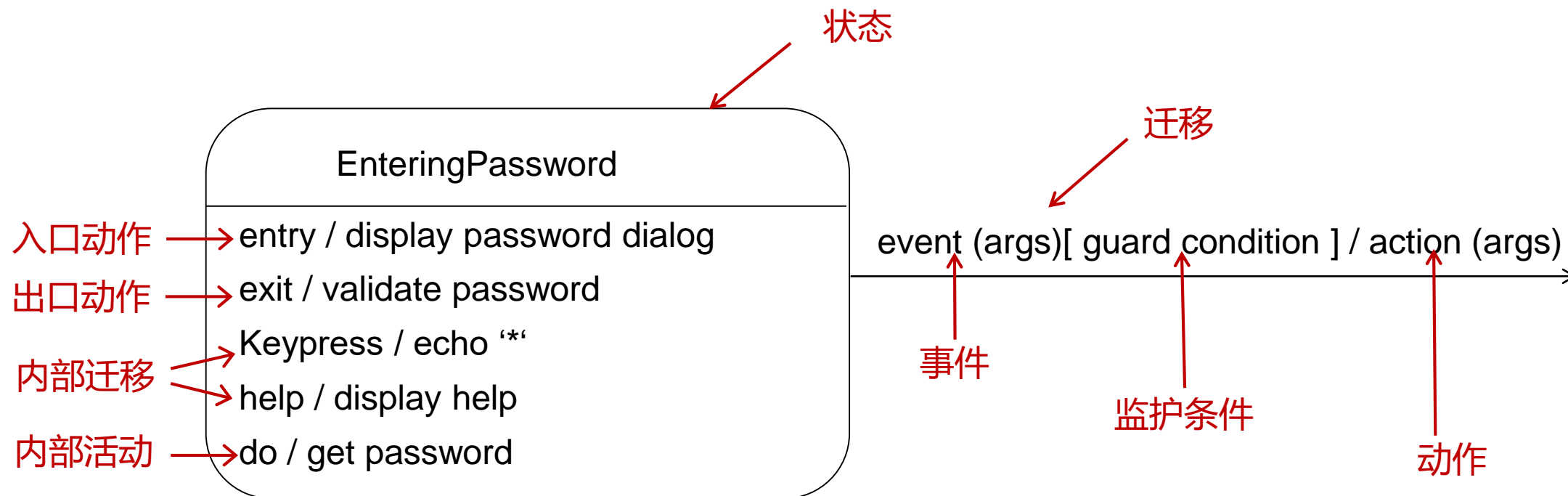


UML 顺序图



UML 状态图

状态图 (Statechart Diagram) 是由状态机扩展而来的，用来描述对象所经过的对外部事件做出相应的状态序列。



UML 状态图

状态 (State)

- 状态是指在对象生命期中的一个条件或状况，对象在此期间将满足某些条件、执行某些活动或等待某些事件。
- UML状态图中定义的状态：初态、终态、中间状态、组合状态、历史状态等。
- 一个状态图只能有一个初态，而终态可以有多个。

事件 (Event)

- 事件是对一个在时间和空间上占有一定位置的有意义事情的规格说明，在状态机的语境中，事件应触发一个状态转换。

UML 状态图

迁移 (Transition)

- 迁移表示两个状态之间的关系，指明对象在第一个状态中执行一定的动作，当特定事件发生或特定条件满足时进入第二个状态。
- 内部迁移是一个未离开该状态的迁移。

动作 (Action)

- 动作是由引起模型状态改变或值的返回的可执行原子计算组成。
- 一个动作的执行时间短且不可中断。

活动 (Activity)

- 活动是驻留在某些状态上的对象将执行的行为。
- 一个活动的执行需要一段时间且可中断，只能在状态内部发生。

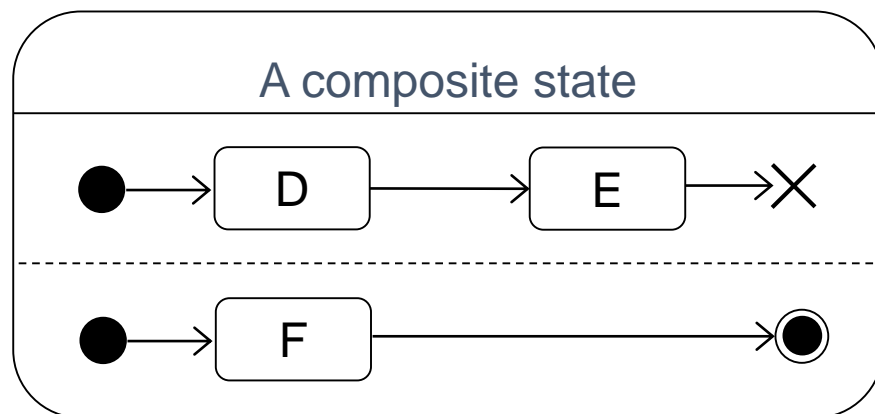
UML 状态图

● 初态：用实心圆点表示

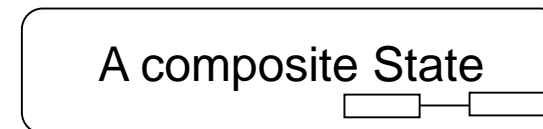
⦿ 终态：用圆形内嵌圆点表示

④ 历史状态是一个伪状态，其目的是记住从组合状态中退出时所处的子状态，当再次进入组合状态时，可直接进入这个子状态，而不是从组合状态的初态开始。

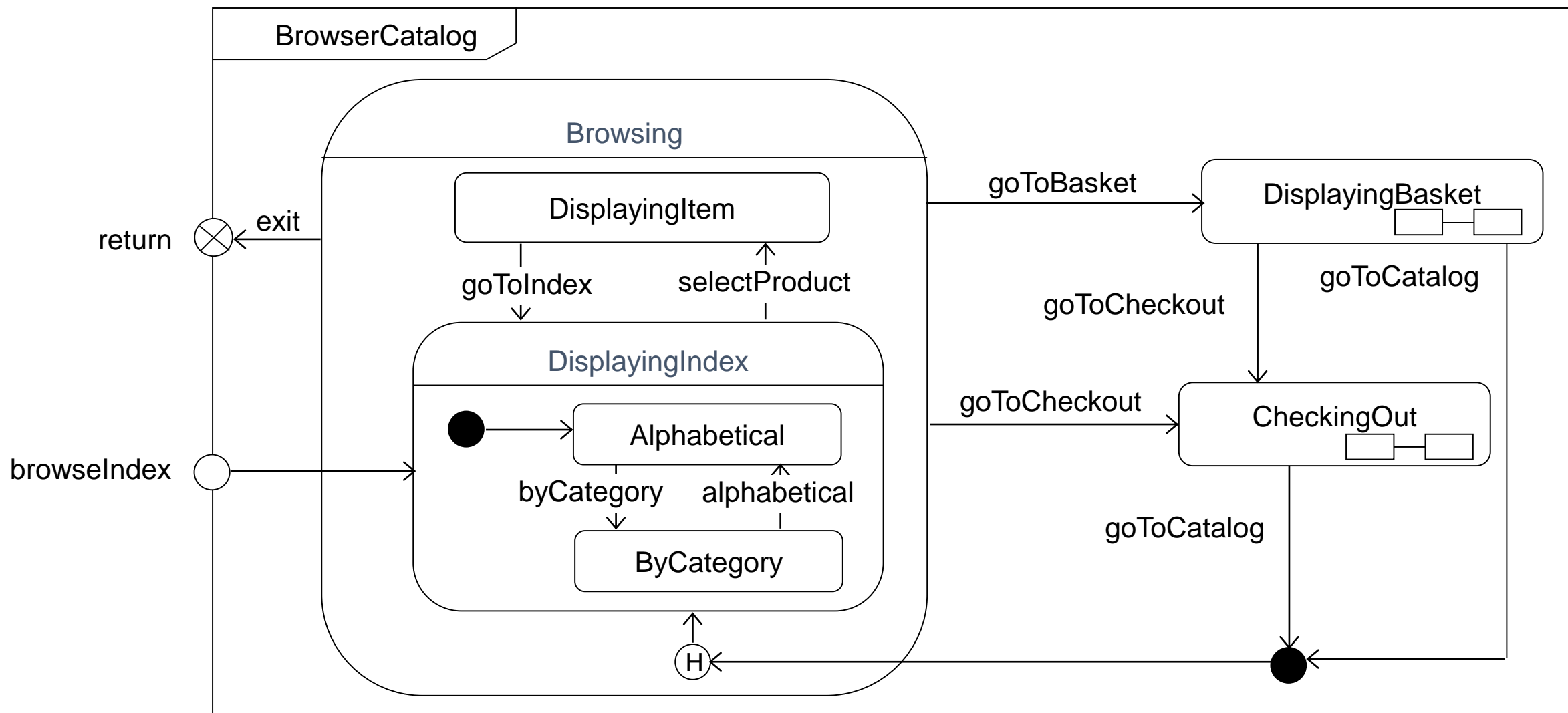
区域



子状态机



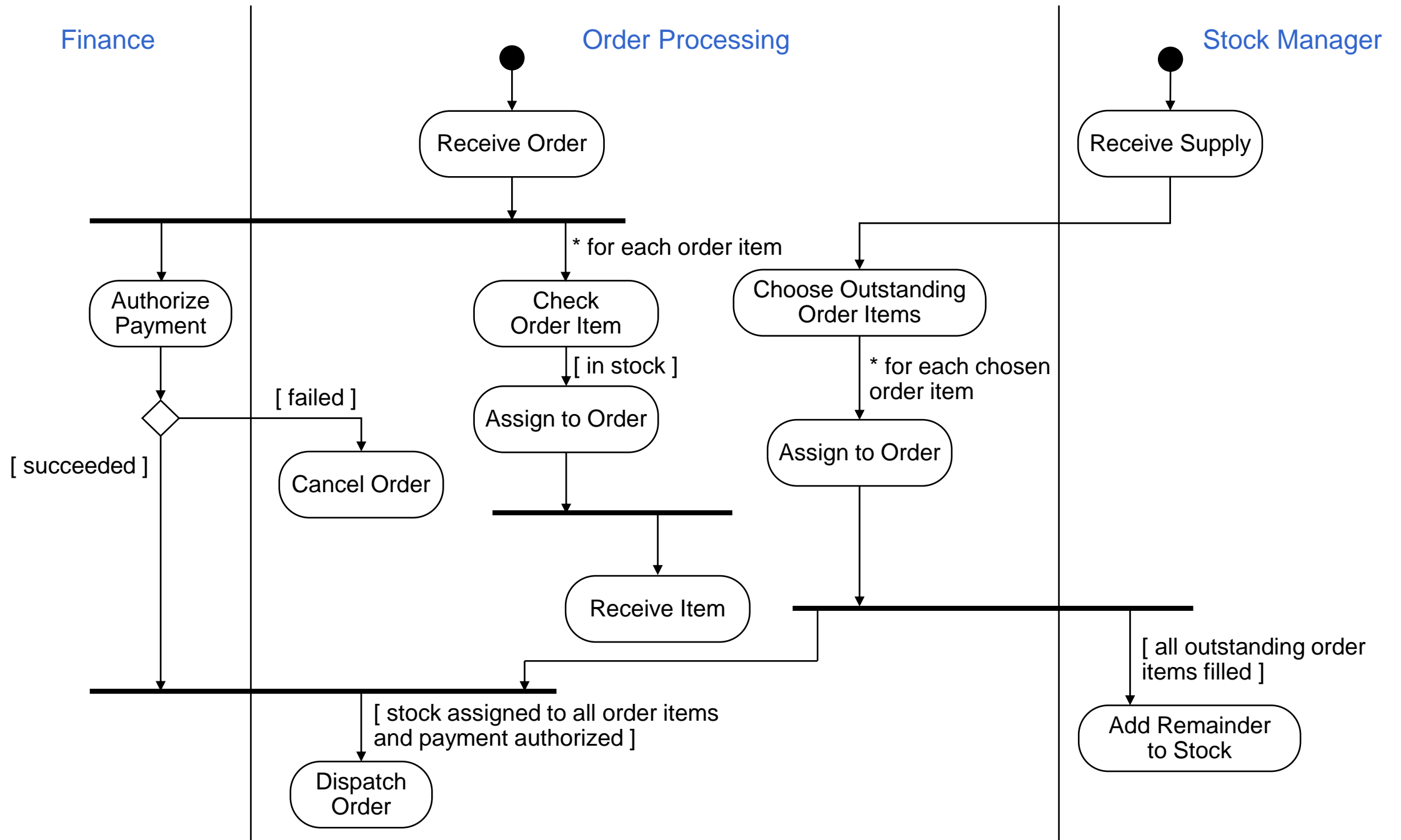
UML 状态图



UML 活动图

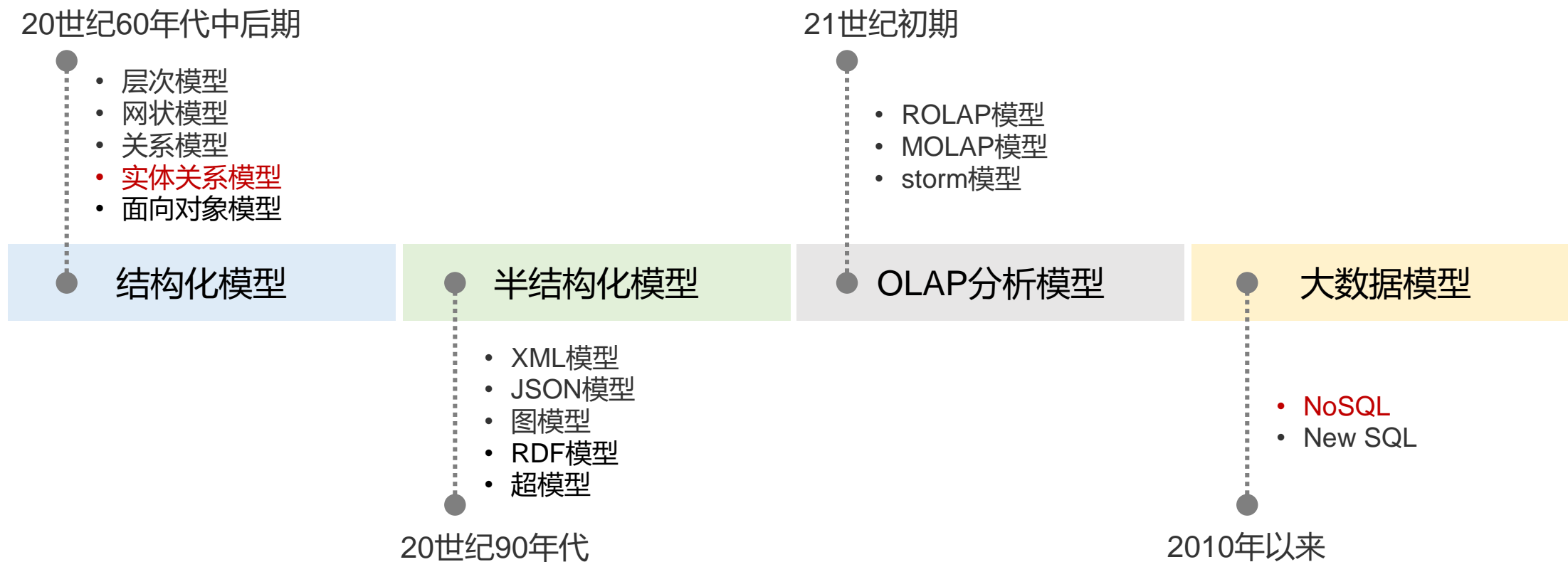
活动图 (Activity Diagram) 本质上是一种流程图，用于描述事物或对象的活动变化流程，是一种表述业务过程或工作流的技术。

- 活动 (Activity)：一个过程中进行的非原子的执行单元。
- 转移 (Transition)：表示活动之间、活动与状态之间的转换。
- 泳道 (Swimlane)：将一个活动图中的活动分组，每一组表示一个特定的类别、人或部门，他们负责完成组内的活动。
- 分支 (Branch)：一般用于表示对象类所具有的条件行为。
- 分叉和汇合 (Fork and Join)：分叉用于将动作流分为两个或者多个并发运行的分支，而汇合则用于同步这些并发分支，以达到共同完成一项事务的目的。
- 对象流 (Object Stream)：动作状态或者活动状态与对象之间的依赖关系。



数据模型

数据模型是现实世界数据特征的抽象，用于描述一组数据的概念和定义。数据模型是数据库中数据的存储方式，是数据库系统的基础。



实体关系模型



实体关系图 (Entity-Relationship Diagram, ERD) 是一种最常用的数据建模工具。



系统出现的不同事物，诸如具体的人或物，或者是抽象概念



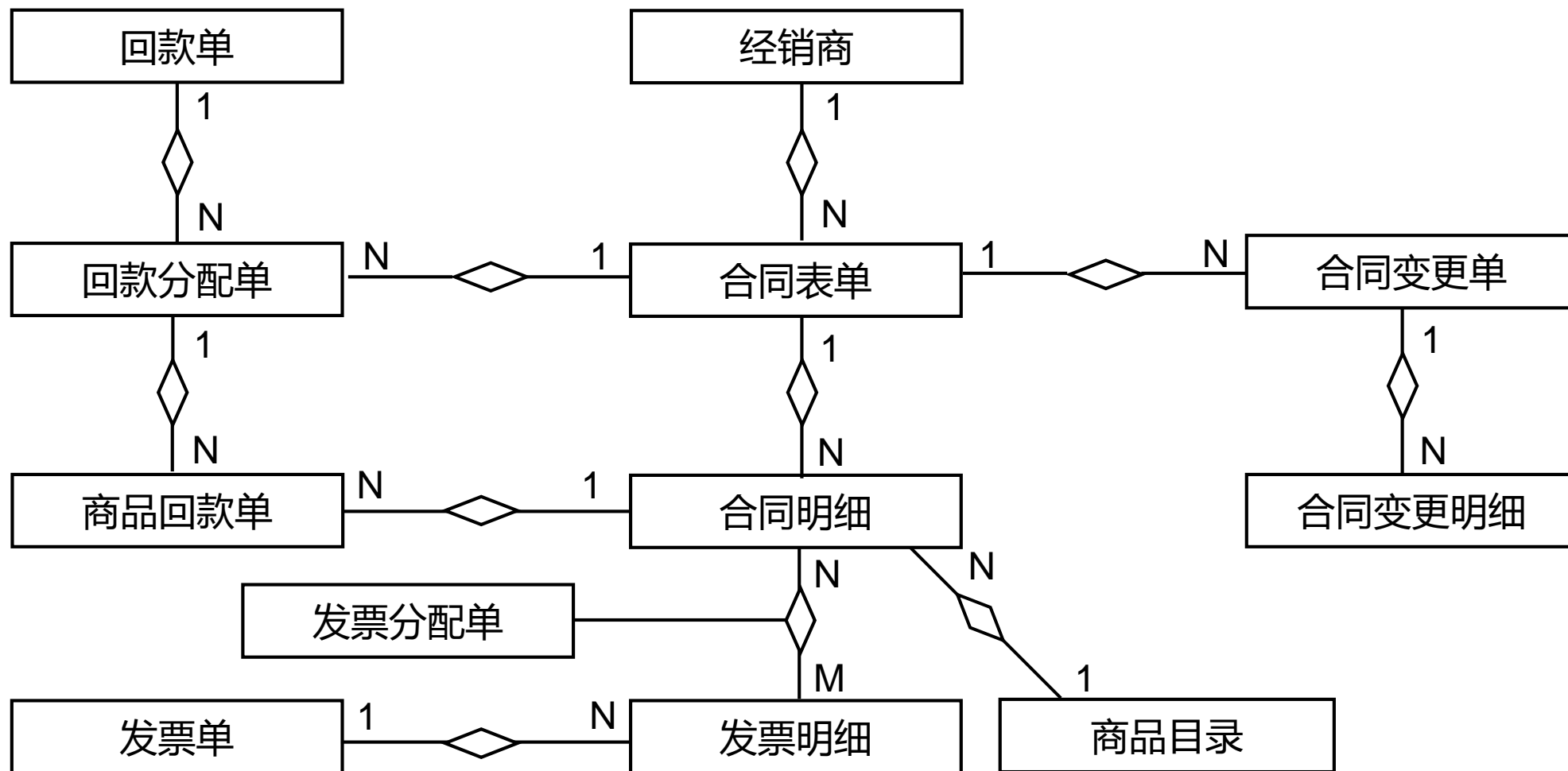
实体所具有的特性，属性是相对实体而言的

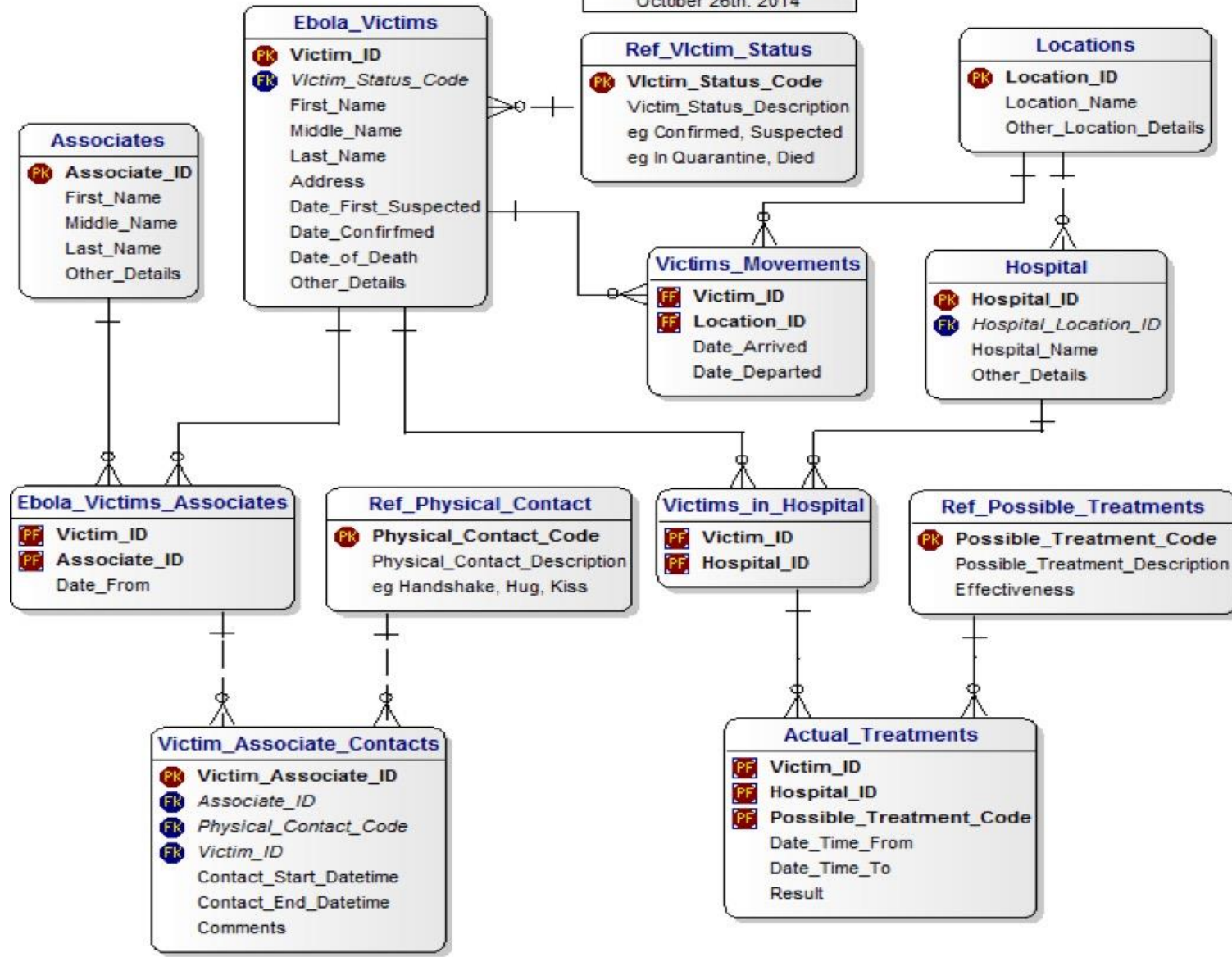


实体之间的相互关联

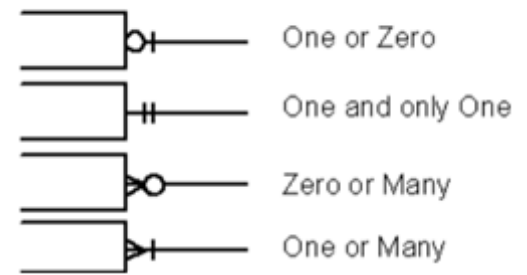
关系基数：在一个给定的关系中实体对象出现的次数 (1:1、1:n、m:n)

实体关系模型





Summary of Crow's Foot Notation



- 主键PK：确定一条记录的唯一标识
- 外键FK：另一个表的主键，用来和其他表建立联系

大数据模型



Key-Value 模型:



- 数据模型：由Key指向Value的键-值对构成，通常用哈希表来实现。
- 应用场景：内容缓存，主要用于处理大量数据的高访问负载，也用于一些日志系统等。
- 优点是查找速度快；缺点是数据无结构化，通常只被当作字符串或二进制数据。

Key-Document 模型:



- 数据模型：Key-Value对应的键值对，Value为结构化数据。
- 应用场景：Web应用，与Key-Value模型相似，不同的是数据库能够了解Value的内容。
- 优点是数据结构要求不严格，表结构可变；缺点是查询性能不高，缺乏统一的查询语法。



1

软件分析建模

2

统一建模语言 UML

3

需求原型与文档化

需求定义

- 该文档列出客户希望实现的所有愿望；
- 该文档应该易于用户理解，一般不采用技术性很强的语言，而是采用自然语言和直观图形相结合的方式进行描述；但自然语言表达容易含糊和不准确。

需求规格说明

- 该文档以规范的方式重述待开发系统应该具有的行为；
- 该文档使用明确的、一致的、全面的表达形式，详细描述软件系统需求，诸如结构化语言、可视化模型、形式化方法等。

需求规格说明



应该包括的内容：

- **功能：** 软件应该提供什么功能？
- **外部接口：** 软件如何与人、系统硬件或者其他软件系统等进行相互作用？
- **性能：** 软件系统在运行速度、可用性、响应时间、恢复时间等方面有什么要求？
- **特性：** 软件系统在可移植性、可维护性、安全性、可靠性等方面有什么考虑？
- **设计约束：** 是否存在标准、开发语言、系统数据库、资源限制、运行环境等因素的影响和策略？

不应包括的内容：

- 诸如项目开发计划、产品保证计划和软件设计细节等内容

需求规格说明

1. 引言

- 1.1 目的
- 1.2 范围
- 1.3 定义、缩写词与简写
- 1.4 参考文献
- 1.5 内容组织

2. 综合描述

- 2.1 产品前景
- 2.2 产品功能
- 2.3 用户特征
- 2.4 一般性限制
- 2.5 假设和依赖

3. 详细需求

- 3.1 功能需求
- 3.2 外部接口需求
- 3.3 性能需求
- 3.4 质量属性
- 3.5 其他需求

4. 支持信息

附录

SRS (IEEE 830 - 1998)

需求描述的质量属性

单个需求描述质量属性：正确性、完整性、无二义性、优先级、可验证性

- 正确性：对系统功能、行为、性能等的描述必须与用户的期望相吻合。
- 优先级：每一项功能需求、特性和用例都必须指定一个实现优先级。
- 完整性：每一项需求必须完整地描述即将交付使用的功能。
- 无二义性：需求的描述对于所有人都只能有一种明确的且统一的解释。
- 可验证性：所描述的需求可以运用一些可行的手段对其进行验证和确认。

需求描述的质量属性

整体需求描述质量属性：完整性、一致性、可修改性、可跟踪性

- **完整性：**应包括软件要完成的全部任务，不能遗漏任何必要的需求信息。
- **一致性：**需求规格说明对各种需求的描述不能存在互相矛盾。
- **可修改性：**文档的格式和组织方式应保证后续的修改能够比较容易和协调一致。
- **可跟踪性：**每一项需求都能与对应的来源、设计、源代码和测试用例等联系起来。

需求描述的质量属性



常见问题：非功能需求的描述是模糊的和不精确的，这样会给需求验证造成困难，引起客户与开发人员之间的争议。

系统应该容易使用，而且使用户的错误降到最少。



需求描述的质量属性

解决方法：使用可验证的量化指标描述非功能需求。

期望目标

系统应该容易使用，而且使用户的错误降到最少。

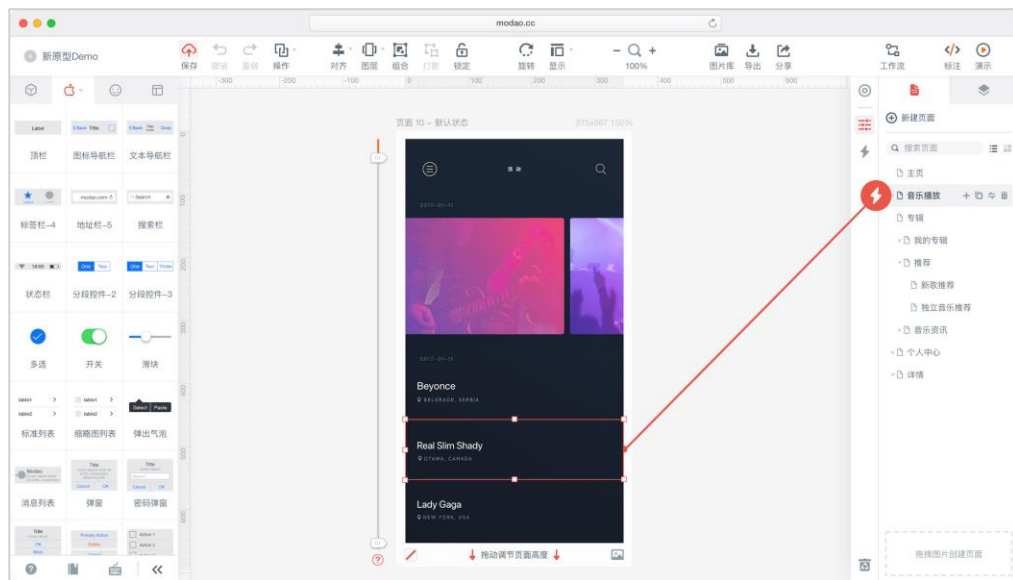


可验证的
非功能需求

对于一个没有经验的用户来说，经过两个小时的培训就应该能够使用系统的所有功能。经过培训后，一个有经验的用户每天的平均出错数不应该超过两次。

需求原型化

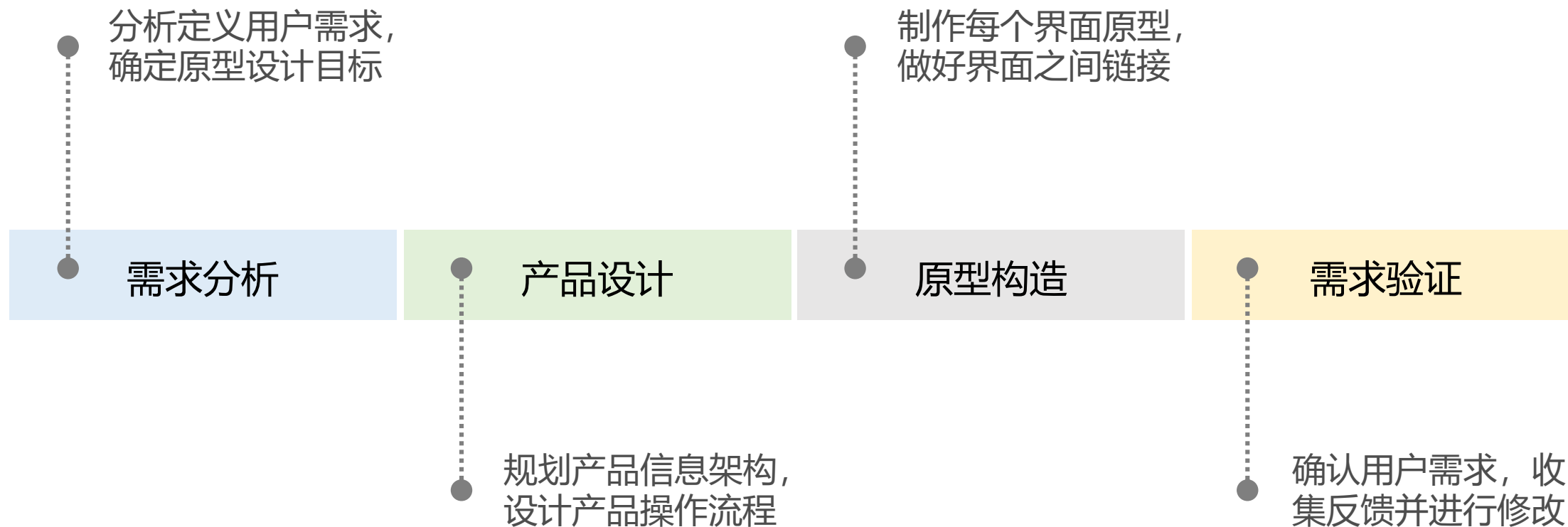
原型法是在实际开发产品之前，先实现该产品的部分原型，并据此征求对需求的反馈意见，它比开发人员常用的技术术语更易于理解。



一张图胜过千言万语!

<https://modao.cc/>

需求原型设计



案例：参观清华

《参观清华》是清华大学校园参观预约系统的官方小程序，为公众提供校园参观预约、热门景点介绍和推荐参观路线等服务，实现校园参观的便捷服务和有序管理。



小程序



人脸识别

2018.7.21 ~ 2019.10.8

总用户数：236.6万人

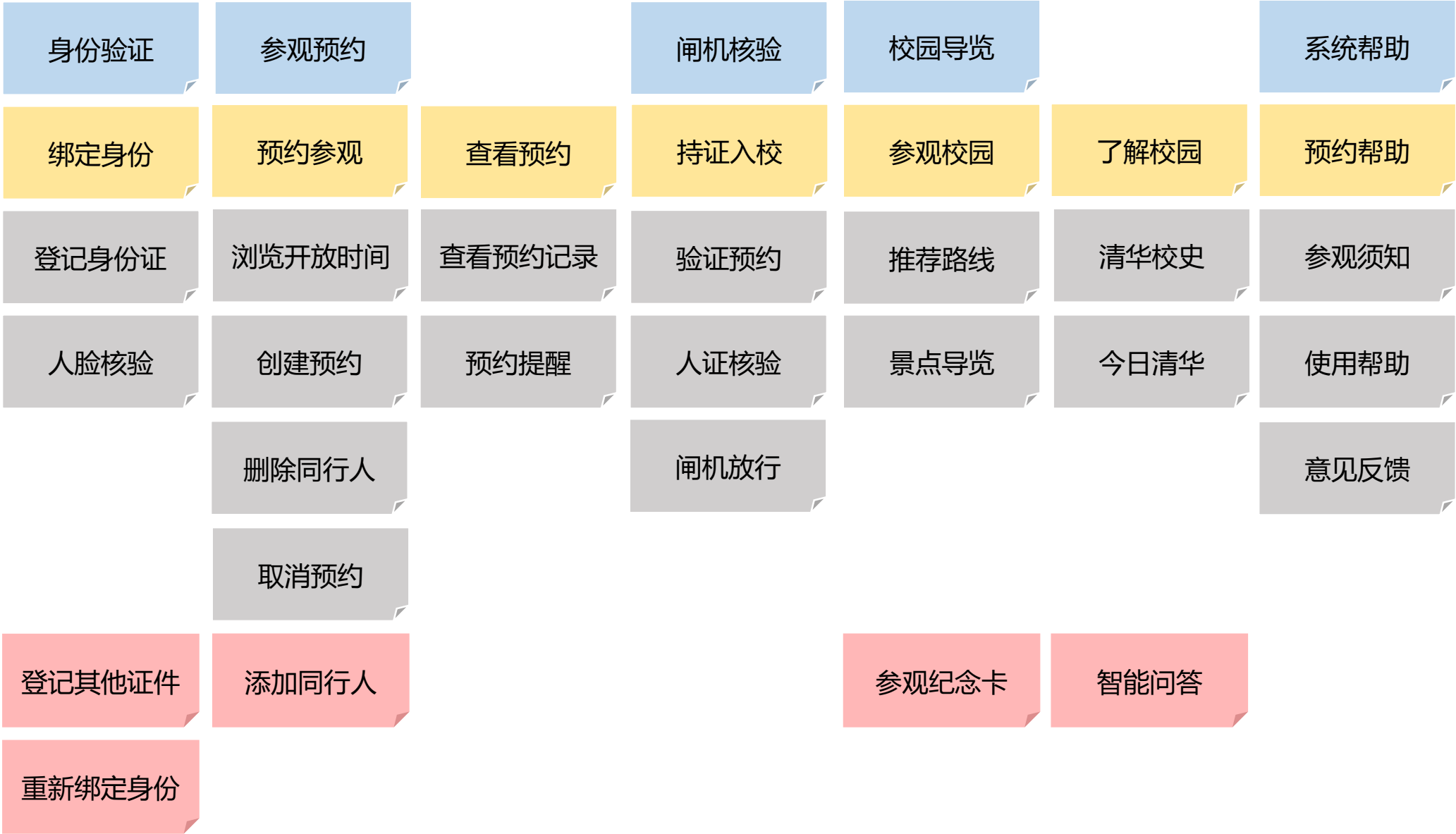
总访问数：3492万次

业务流程（时间线）

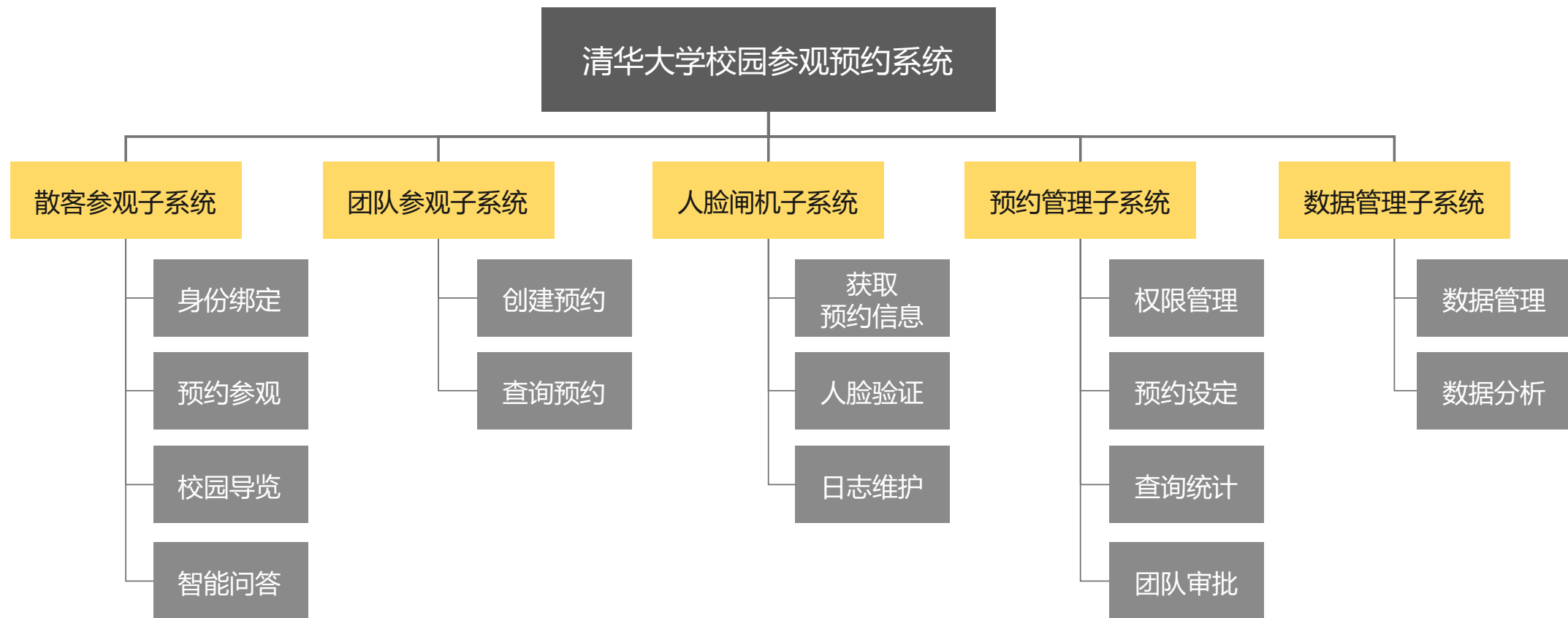
高

商业价值

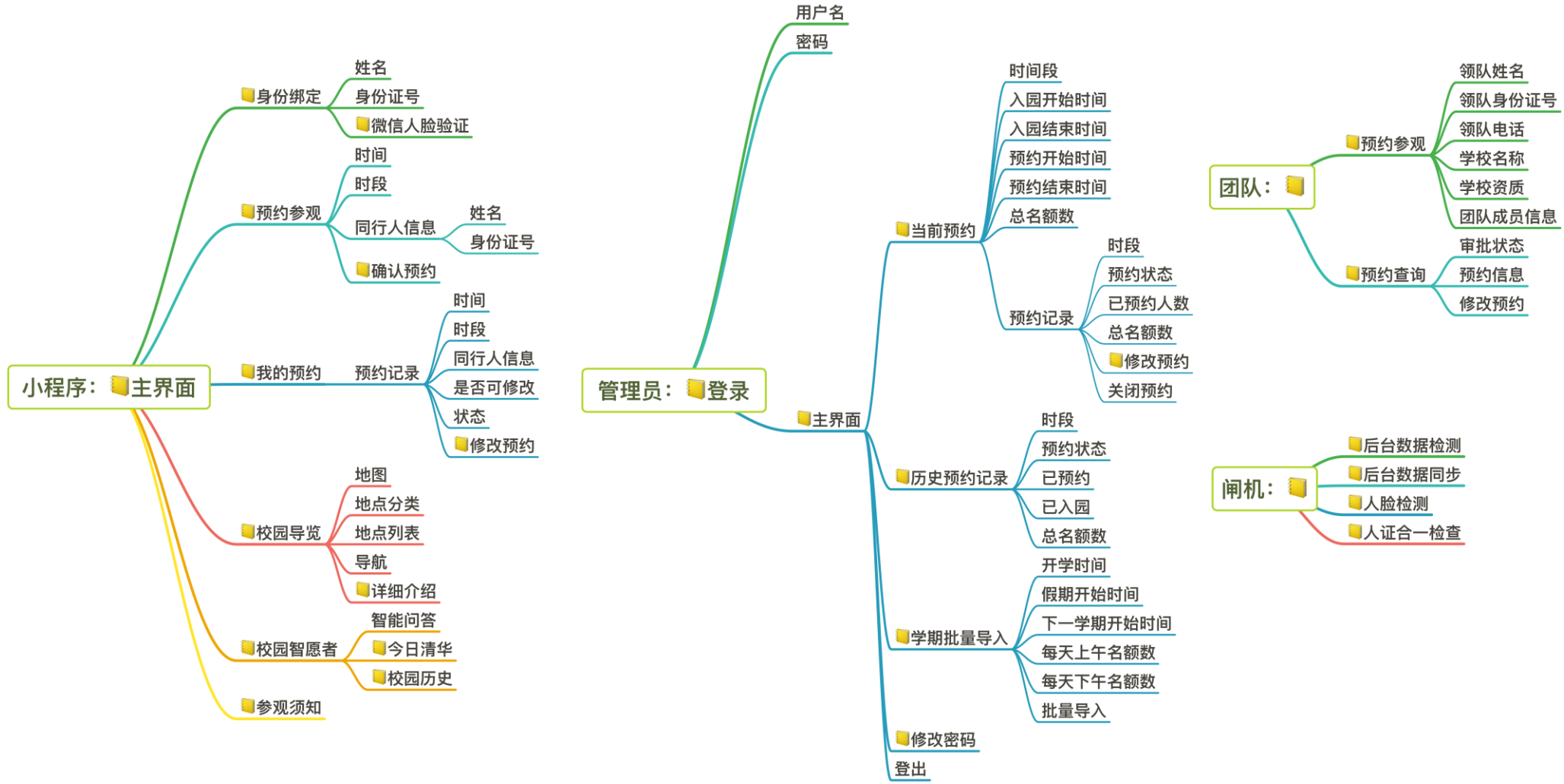
低



功能模块划分



产品信息结构图





预约参观页面：

- 显示当日开始之后八天的状态
- 状态：关闭、开放、已满
- 点击可预约日期，显示可入校时段
- 时段：9-10，10-11，14-15，15-16
- 点击同行按钮，允许输入同行人的信息
- 每个预约人最多可有两个同行人的信息
- 在选择日期和时段后，点击预约才进入确认预约的页面



添加同行人的页面：

- 验证同行人的身份证号规则，但不进行姓名和身份证号匹配验证
- 验证当前页面的预约人和同行人的证件号是否重复

原型设计工具



墨刀：交互原型设计，梳理页面结构和交互逻辑

<https://modao.cc/>



Sketch：UI设计，像素级界面细节

<https://www.sketch.com/>



蓝湖：产品设计写作，看标注，下切图

<https://lanhuapp.com/>



谢谢大家!

THANKS

