



# 第四章 需求分析



# 课程内容

- **4.1 需求分析的任务**
- 4.2 建立系统关联图
- 4.3 分析需求的可行性
- 4.4 构建用户接口原型
- 4.5 确定需求的优先级
- 4.6 需求建模
- 4.7 建立数据词典



## 4.1 需求分析的任务

- **需求分析**是需求工程中最核心的工作，**需求建模**是需求分析的主要手段
- **基本任务**：分析和综合已收集到的需求信息
  - **分析**：找出需求信息间的内在联系和可能的矛盾
  - **综合**：去掉那些非本质的信息，找出解决矛盾的方法并建立系统的逻辑模型

提炼、分析和仔细审查已收集到的需求信息，找出真正的和具体的需求



## 需求分析的任务(续)

- 需求分析不是分析如何实现用户的需求。实际上，需求分析是以业务分析为导向，将用户零散的需求串联起来，形成一个体系完整、组织合理、内容清晰的框架。

What to do? YES

How to do? NO



## 需求分析的任务(续)

- 需求分析是一个**分解、提炼**的过程，在此过程中**消除**需求矛盾

### 1. 分解

**分解**是人类控制复杂性，认识复杂事物的基本策略方法。无论是采用结构化方法，还是采用面向对象方法，分解都是必须采用的手段。

传统方法一般采用**系统导向**的分解方法，而现代需求工程建议采用**业务导向**的方法。



# 需求分析的任务(续)

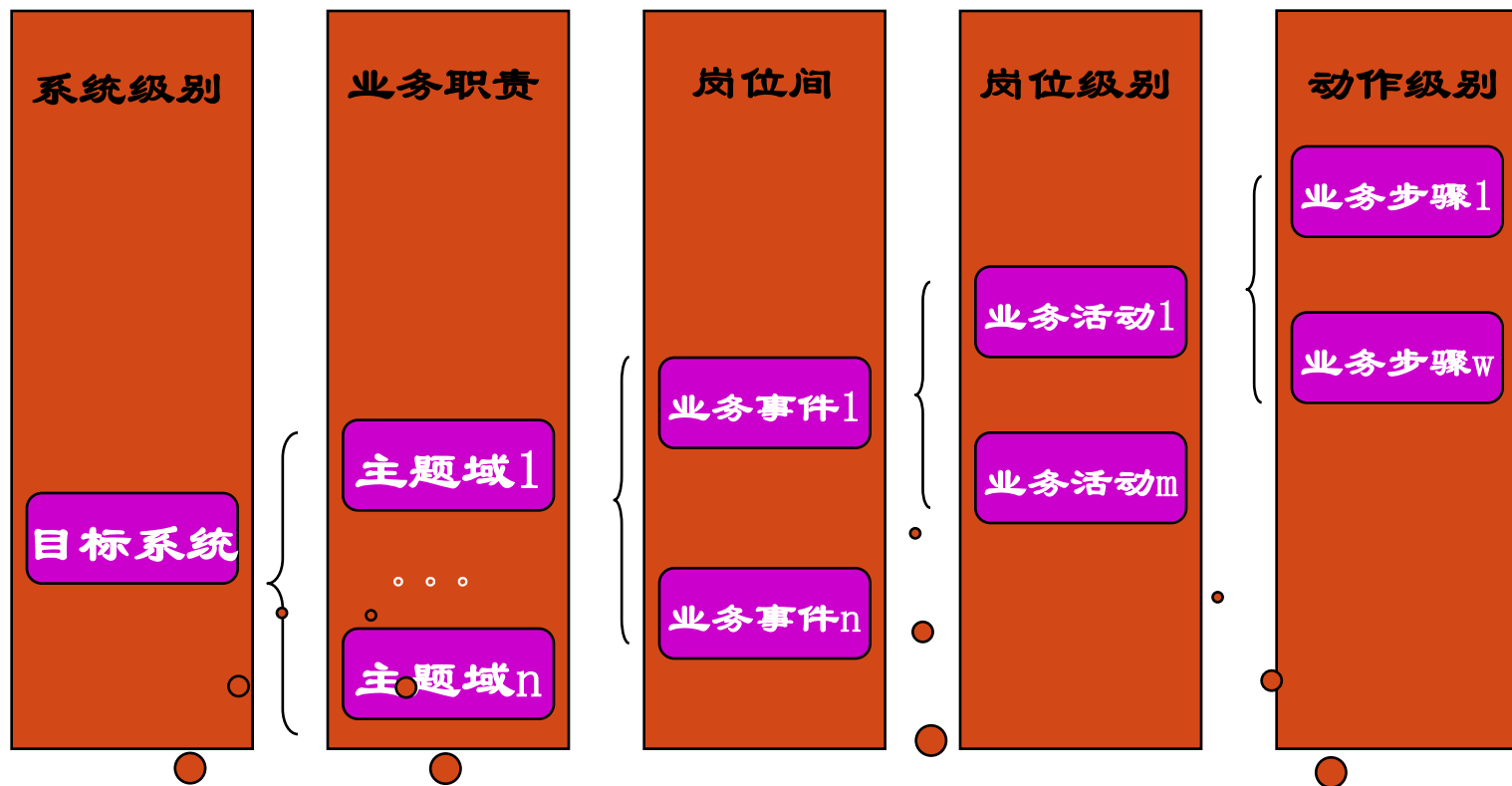
## 1. 分解

### ■ 分解的策略

- 业务流程为主线的分解策略
- 程序结构为主线的分解策略
- 基于场景的分解策略
- 基于数据的分解策略



## 1) 业务流程为主线的分解策略



目标决定  
范围

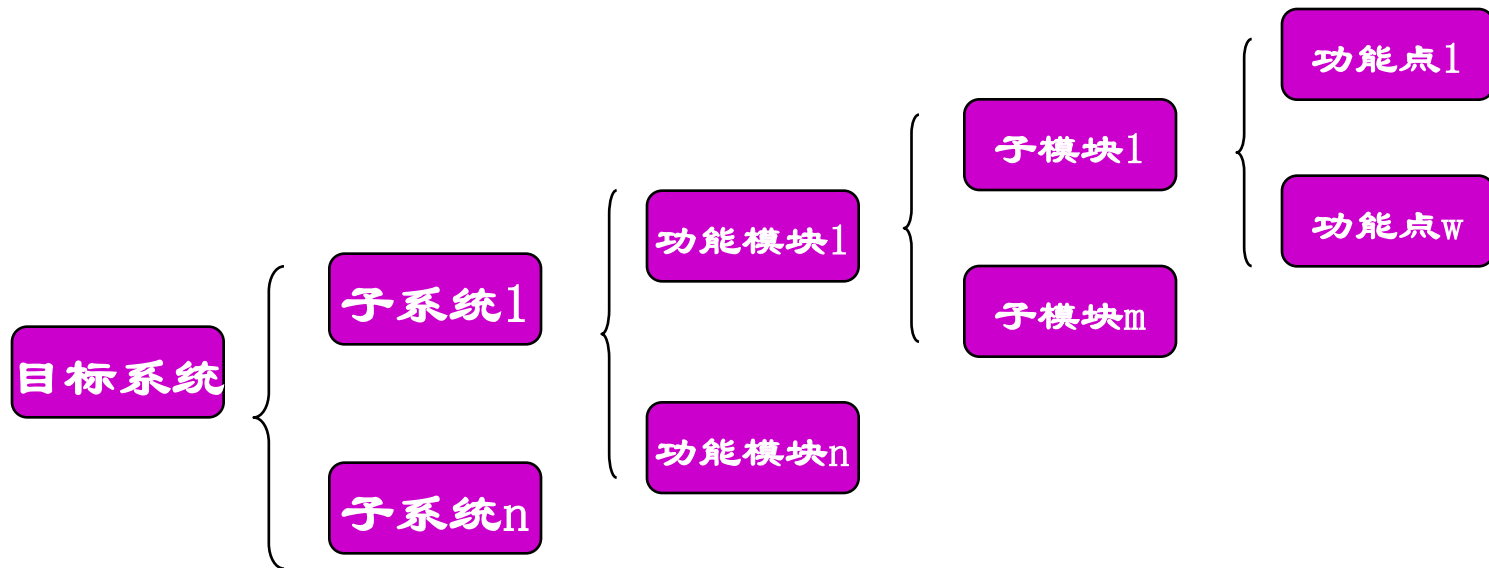
理清业务  
脉络

填充细节

细化和确认  
工作



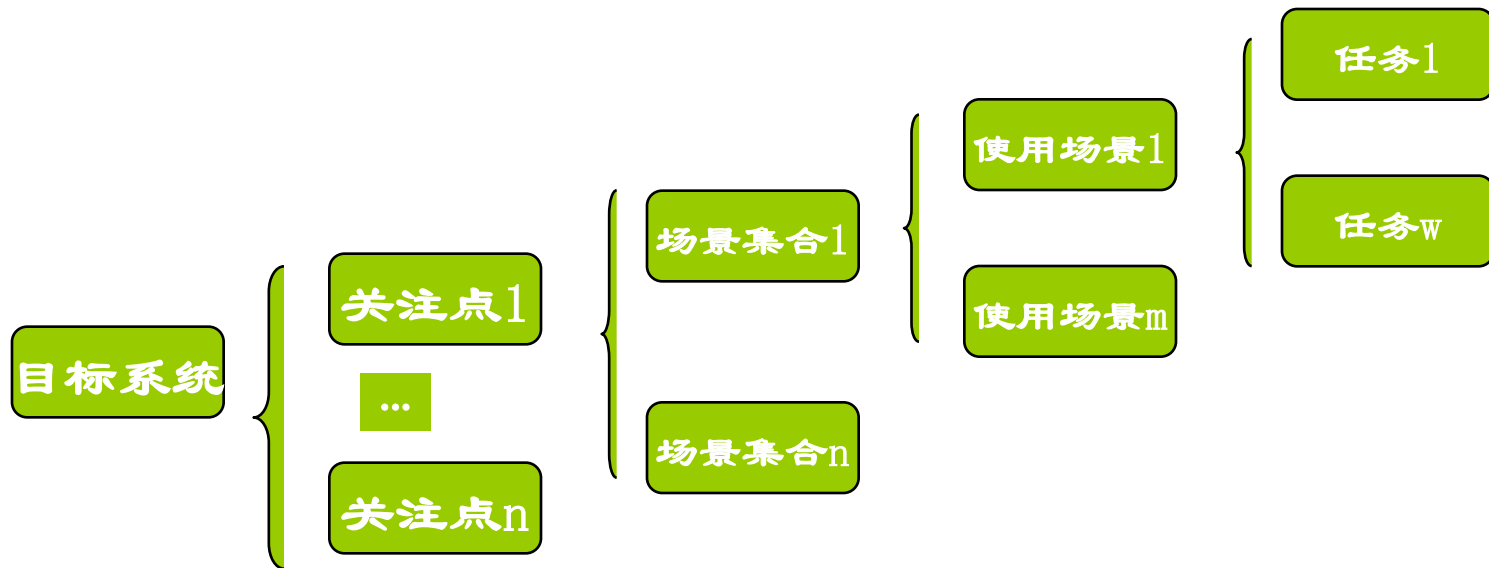
## 程序结构为主线的分解策略





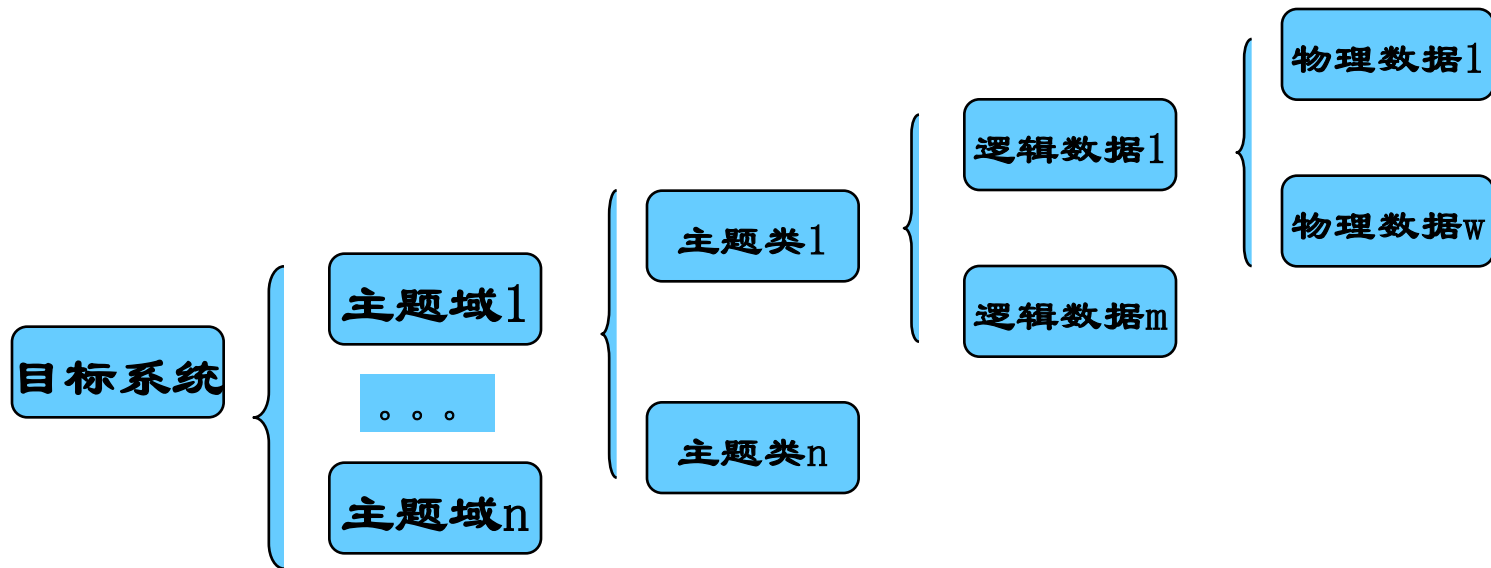


## 基于场景的分解策略





## 基于数据的分解策略





## 需求分析的任务(续)

### 2. 提炼

- 当按照任何一种线索自顶向下进行分解时。就会破坏其它线索的完整性。

例如，如果以“业务”为线索，就会发现数据需求分解后会出现相互交叠的情况，也就是在多个业务事件中都涉及相同的类。

- 这种情况可能会影响需求分析人员建立全面的理解，因此需要采用**自底向上**的方法进行**提炼**。

例如将每个业务事件中的类进行提炼，抽取出共性的部分，建立针对整个系统的全局领域模型。



## 需求分析的任务(续)

### 3. 消除矛盾

在分析过程中，可能会发现**有些需求是相互矛盾的、冲突的**，由于是将收集的信息放在一个预先定义的结构中发现这些矛盾的，因此对矛盾的影响范围会有直观的了解，也能够知道它影响那些层面。寻找相应的人员，通过进一步需求获取来消除矛盾。



## 需求分析的任务(续)

### ■ 需求分析的具体工作：

- 建立系统关联图
- 分析需求的可行性
- 构建用户接口原型
- 确定需求的优先级
- 需求建模
- 建立数据词典



## 课程内容

- 4.1 需求分析的任务
- **4.2 建立系统关联图**
- 4.3 分析需求的可行性
- 4.4 构建用户接口原型
- 4.5 确定需求的优先级
- 4.6 需求建模
- 4.7 建立数据词典



## 4.2 建立系统关联图

### ■ 关联图

用于描述系统与外部实体间的界限和接口的模型，也明确通过接口的信息流和物质流

### ■ 目的

根据需求获取阶段确定的系统范围，用图形表示系统与外部实体间的关联



## 建立系统关联图(续)

### ■ 关联图的画法

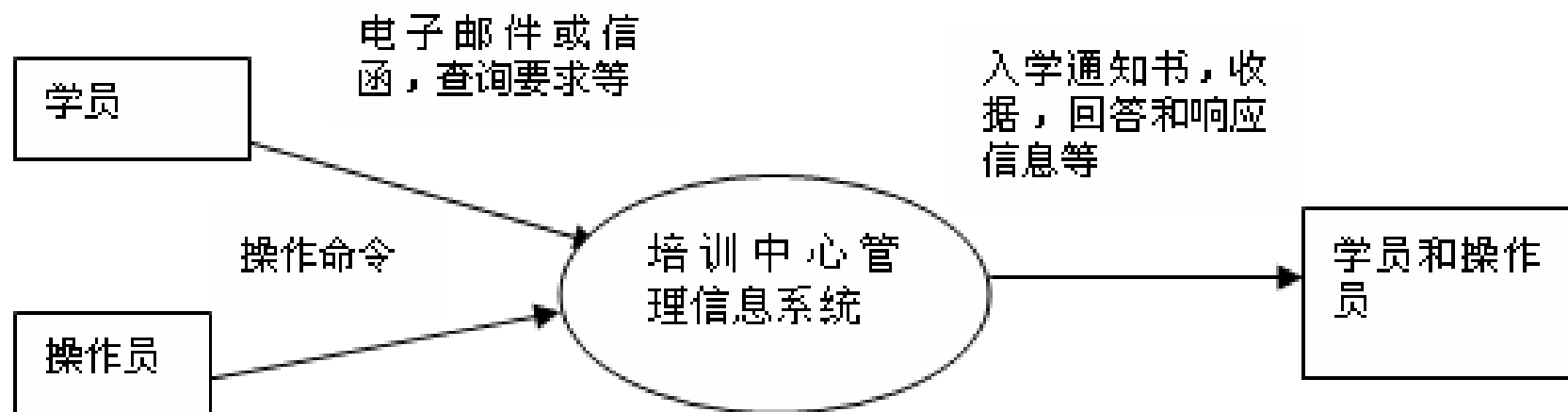
- 用椭圆表示整个要开发的系统，椭圆内标识该系统的名字
- 用方框表示系统外部实体
- 用带标识的有向边表示系统与外部实体间的关系和信息流向





## 建立系统关联图(续)

### ■ 例：培训中心管理信息系统的关联图





## 建立系统关联图(续)

### ■ 建立系统关联图的好处

- 项目相关人员一开始不必考虑太多的细节,而是把注意力集中在软件系统的接口方面
- 确定系统的界限
- 为分析用户需求提供很好的依据
- 使项目相关人员更易于理解和审查



## 课程内容

- 4.1 需求分析的任务
- 4.2 建立系统关联图
- **4.3 分析需求的可行性**
- 4.4 构建用户接口原型
- 4.5 确定需求的优先级
- 4.6 需求建模
- 4.7 建立数据词典



## 4.3 分析需求的可行性

### ■ 基本任务

在允许的成本和性能要求以及系统的范围内，分析每项需求得以实施的可能性

### ■ 工作目的

明确与每项需求相关联的风险



## 分析需求的可行性(续)

- 与高风险相关的需求最可能导致软件开发的失败
  - 性能风险：实现这项需求可能导致整个系统性能的下降
  - 安全风险：可能导致无法满足整个系统的安全需求
  - 过程风险：可能导致需要对常规的开发过程做修改
  - 实现技术风险：可能需要使用不熟悉的实现技术



## 分析需求的可行性(续)

- 与高风险相关的需求最可能导致软件开发的失败
  - 数据库风险：可能导致系统不支持的非标准数据
  - 日程风险：可能遇到技术困难并危及系统原定的开发日程
  - 外部接口风险：可能涉及外部接口
  - 稳定风险：这项需求可能是易变的



## 分析需求的可行性(续)

- 分析需求可行性的要点：
  - 需要与有经验的开发人员共同分析
  - 需要判断每一项需求的风险
  - 通常使用定性的方法来评估风险



## 课程内容

- 4.1 需求分析的任务
- 4.2 建立系统关联图
- 4.3 分析需求的可行性
- **4.4 构建用户接口原型**
- 4.5 确定需求的优先级
- 4.6 需求建模
- 4.7 建立数据词典





## 4.4 构建用户接口原型

### ■ 原型

如果在最终的物件 (final artifact) 产生之前, 一个中间物件 (mediate artifact) 被用来在一定广度和深度范围内表现这个最终物件, 那么这个中间物件就被认为是最终物件在该广度和深度上的原型

### ■ 用户接口原型

一个可能的局部实现, 而不是整个系统



## 构建用户接口原型(续)

■ 例如：

某国家机关对其开发的系统在招标时要求，应标系统除了完成必要的商务标和技术标应答外，还需要根据其提供的用例完成相关系统的开发，其中该系统大约25-30个用例，招标时提供的用例为8个。



## 构建用户接口原型(续)

### ■ 目的

澄清一些不能确定的或含糊的需求，尽早使这些需求能完整和清除地表达出来

### ■ 基本任务

对于软件开发人员或用户不能明确化的需求，通过建立相应的用户接口原型然后评估该原型，使得项目相关人员能更好理解所要解决的问题。



## 构建用户接口原型(续)

### ■ 构造原型前需确定类型

- 抛弃型原型：在原型达到预期目的后将其抛弃
- 进化型原型：在需求清楚定义的情况下，以渐增式方式构建原型，并使原型最终能成为软件产品的一部分



## 构建用户接口原型(续)

### ■ 抛弃型原型

- 花费最小的代价，争取最快的速度
- 可能会使用简易的开发工具和成熟的构造技术
- 可能会忽略或简化处理原型目的不相关的功能特征
- 对不确定的需求是最适合的方法
- 要坚决的抛弃



## 构建用户接口原型(续)

### ■ 进化型原型

- 质量一开始就必须达到最终系统的要求
- 时间多、代价高
- 要易于进行扩展和频繁改进，因此开发者必须重视进化式原型的设计
- 适用于处理清晰的需求、规格说明和技术方案



## 构建用户接口原型(续)

### ■ 构造原型的方法

#### ● 纸上原型化方法

通过书面材料以场景的形式呈现给用户、代价小

#### ● 人工模拟原型化方法

根据用户的输入由人模拟系统的响应、代价较小

#### ● 自动原型化方法

用开发语言、开发环境开发一个可执行的原型、成本较高



## 课程内容

- 4.1 需求分析的任务
- 4.2 建立系统关联图
- 4.3 分析需求的可行性
- 4.4 构建用户接口原型
- **4.5 确定需求的优先级**
- 4.6 需求建模
- 4.7 建立数据词典





## 4.5 确定需求的优先级

### ■ 好处

- ① 帮助项目相关人员判断系统的核心需求，并有助于项目相关人员集中于重点问题的交流和协商；
- ② 需求优先级之间的关联可以帮助软件开发人员决定软件体系结构，还可以帮助解决可能发生的设计冲突；
- ③ 根据需求的优先级权衡合理的项目范围和进度安排、预算、人力资源以及质量目标的要求。



## 确定需求的优先级(续)

- 理想情况：需求获取阶段由客户决定需求的优先级
- 实际情况：优先级的分配应由软件开发人员和项目相关人员共同完成，最好是在做了一些初始的分析工作后，再进行需求优先级的分配



## 确定需求的优先级(续)

- 重要性：需求的不可或缺程度
- 紧急性：需求的时间紧迫程度
- 惩罚性：忽略需求会导致的惩罚程度
- 成本：实现需求的代价
- 风险：需求实现中可能产生的风险程度



## 确定需求的优先级(续)

命 名	含 义	方法来源
高	一个关键任务的需求或下一版本所需要的。	Karl E. W.
中	支持必要的系统操作或最终所要求的，但如果有必要的话，可以延迟到下一个版本。	
低	功能或质量上的增强；如果资源允许的话，实现这些需求总有一天使产品更完美。	
基本的	只有在这些需求上达成一致意见，软件才会被接受。	IEEE 1998
条件的	实现这些需求将增强产品的性能，但如果忽略这些需求，产品也是可以被接受的。	
可选的	对一个功能类有影响，实现或不实现均可。	
3	必须完美地实现。	Kovize 1999
2	需要付出努，但不必做得太完美。	
1	可以包含缺陷	



## 确定需求的优先级(续)

- 在确定了所有需求的优先级后，必须把每个需求优先级记录到需求规格说明中，并可通过设置相应的字段给予标识。



## 课程内容

- 4.1 需求分析的任务
- 4.2 建立系统关联图
- 4.3 分析需求的可行性
- 4.4 构建用户接口原型
- 4.5 确定需求的优先级
- **4.6 需求建模**
- 4.7 建立数据词典



## 4.6 需求建模

- 建模是需求分析的主要手段，它通过简化（化简）、强调来帮助需求人员理清思路，达成共识。
- 模型
  - 为了理解事物而对事物做出的一种抽象，是对事物的一种无歧义的书面描述
  - 通常可由文本、图形符号或数学符号以及组织这些符号的规则组成



## 需求建模(续)

### ■ 基本任务

导出目标系统的逻辑模型（或需求模型），以明确目标系统“做什么”的问题。

### ■ 目的

- 和用户达成对信息内容的共同理解
- 增强对用自然语言描述的需求规格说明的理解，而不是替换它





## 需求建模(续)

### ■ 建模方法：抽象

从不同的角度抽象目标系统的特性，使用精确的方法构造系统的模型

- 一方面要求人们只关注重要的信息，忽略次要的内容

通过强调本质的特征，减少问题的复杂性

- 另一方面也要求人们将认知保留在适当的层次，屏蔽更深层次的细节



## 需求建模(续)

### ■ 需求建模

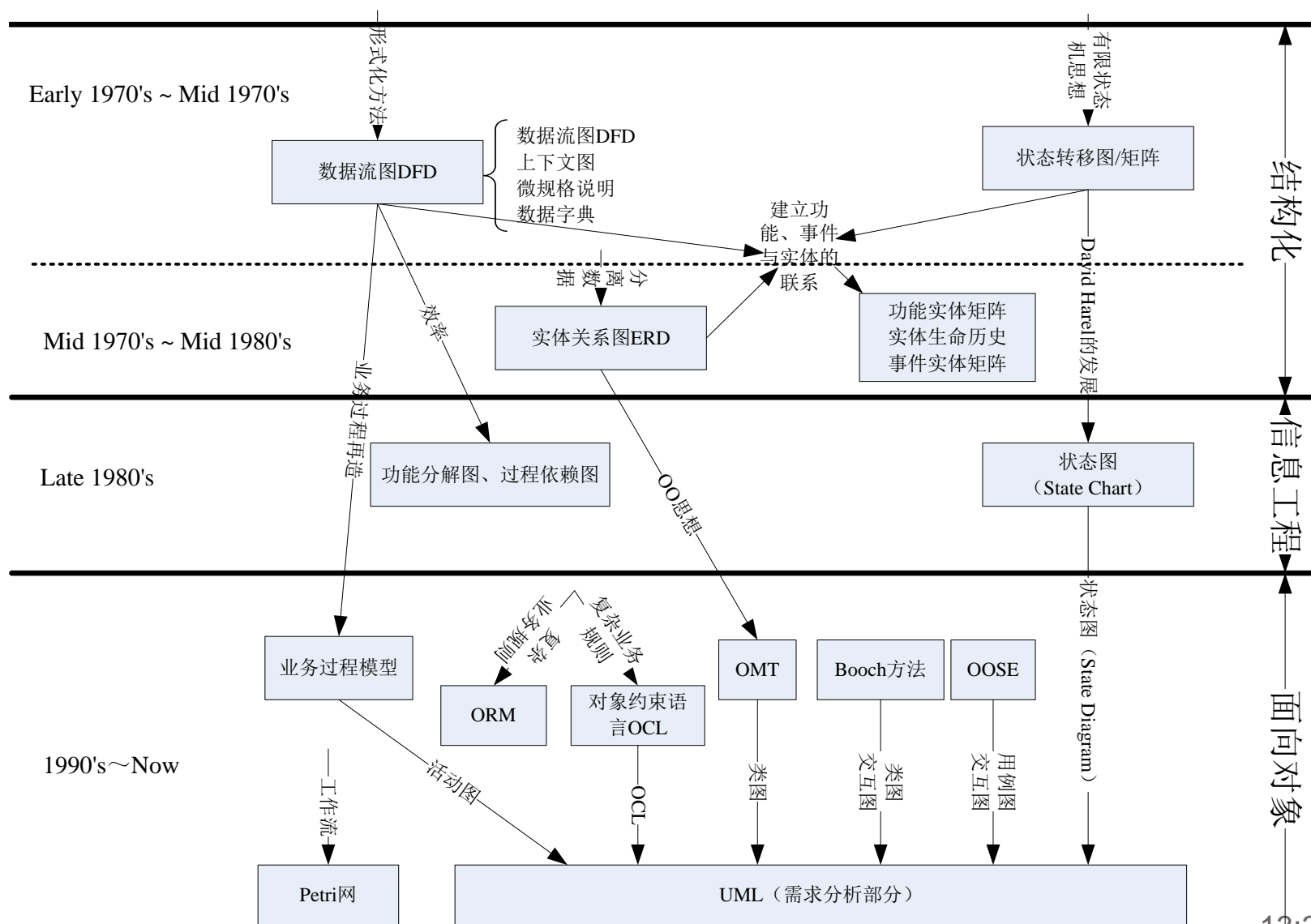
把由文本表示的需求和由图形或数字符号表示的需求结合起来，绘制出对目标系统的完整性描述，以检测软件需求的一致性、完整性和错误等。



## 需求建模(续)

### ■ 需求建模的通常做法：

- 先依据获取的问题域信息建立初步的模型。
- 然后分析用户需求，对模型进行调整，得到一个中间形式的模型形式。
- 最后，对调整后的模型进行逻辑推理和验证，如果符合预期的期望，那么它就是最终的解决方案模型。





❖ 结构化技术

- 数据建模
  - 实体关系图Entity Relationship Diagram
- 过程建模
  - 数据流图Data Flow Diagram
  - 上下文图Context Diagram
  - 微规格说明Mini-Specification
  - 数据字典Data Dictionary
- 行为建模
  - 状态(转换)图/矩阵State (Transition) Diagram/Matrix
- 过程/数据关系建模
  - 功能实体矩阵Function/Entity Matrix
- 信息工程方法
  - 功能分解图Function Decomposition Diagram
  - 过程依赖图Process Dependency Diagram

■ 面向对象技术

□ UML

- 用例图Use-Case Diagram
- 类图Class Diagram
- 交互图（顺序图/通信图）Interaction (Sequence / Communication) Diagram
- 活动图Activity Diagram
- 对象约束语言Object Constraint Language
- 状态图State Chart Diagram



## 需求建模(续)

- 半形式化的建模

图形表示的需求分析方法

- 形式化的建模

VDM、Z符合、B方法、基于代数理论的方法



## 课程内容

- 4.1 需求分析的任务
- 4.2 建立系统关联图
- 4.3 分析需求的可行性
- 4.4 构建用户接口原型
- 4.5 确定需求的优先级
- 4.6 需求建模
- **4.7 建立数据词典**



## 4.7 建立数据词典

### ■ 数据词典

定义目标系统中使用的所有数据元素和结构的含义、类型、数量值、格式和度量单位、精度及允许取值范围的共享数据仓库。

### ■ 作用

确保软件开发人员使用统一的数据定义，以及可提高需求分析，设计、实现和维护过程中的可跟踪性。





## 建立数据词典 (续)

- 每个项目建立一个独立的数据词典
- 每个数据项对应数据词典中的一项纪录，可使用简单的符号定义

数据项名 = 数据项定义

数据项定义 = 数据类型 + 数量值 + 数量单位  
+ 允许的取值范围 + ...