

第6章.需求分析概述

韩锐

北京理工大学 计算机学院

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

主要内容

1. 需求分析的根本任务
 1. 建立分析模型
 2. 建立解决方案
2. 需求分析技术
3. 需求分析方法
4. 前期需求阶段的建模与分析
5. 需求分析的活动

1. 需求分析的根本任务

获取结果

- 用户的理解
- 问题的描述

需求分析

- 建立分析模型
- 创建解决方案

需求开发目标

- 共同的理解
- 解决方案的描述

1. 需求分析的根本任务

■ 1.1. 建立分析模型

- 将复杂的系统分解成为简单的部分以及它们之间的联系，确定本质特征
- 和用户达成对信息内容的共同理解
- 分析的活动主要包括识别、定义和结构化，它的目的是获取某个可以转换为知识的事物的信息

1. 需求分析的根本任务

■ 1.2. 创建解决方案

- 将一个问题分解成独立的、更简单和易于管理的子问题来帮助寻找解决方案
- 创建解决方案的过程是创造性的
- 帮助开发者建立问题的定义，并确定被定义的事物之间的逻辑关系
 - 这些逻辑关系可以形成信息的推理，进而可以被用来验证解决方案的正确性。

1.1. 建立分析模型

■ 模型

- **定义：**“模型是对事物的抽象，帮助人们在创建一个事物之前可以有更好的理解”
- 集中关注问题的计算特性（数据、功能、规则等等）
- “它是对系统进行思考和推理的一种方式。建模的目标是建立系统的一个表示，这个表示以**精确一致的方式**描述系统，使得系统的使用更加容易”
- 建模方法
 - 抽象
 - 分解
 - 投影

1.1. 建立分析模型：建模方法

■ 抽象（Abstraction）

- 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，就减少了问题的复杂性
- 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
 - 在问题的各元素之间推断出更广泛和更普遍的关系，帮助人们寻找解决方案

■ 分解（Decomposition / Partitioning）

- “分而治之”
 - 将单个复杂和难以理解的问题分解成多个相对更容易的子问题，并掌握各子问题之间的联系
- 分解的方案往往还能提供问题的解决思路

■ 投影（Projection）

- 多视点方法

抽象案例

- 以做饭为例
- 洗菜 --> 炒菜
- 做饭这件事情实际规定为了两个活动，包括**洗菜**活动，**炒菜**活动，并且指定了先后次序。
- 你不会从中得到具体的洗菜、炒菜细节上的指导，但你却能够明白应该先进行洗菜然后炒菜，以及自己处在哪个活动阶段。

分解案例

- 以网购需求为例
- APP中要展示商品列表
- 用户能够将商品添加至购物车，并进入购物车结算
- 用户能够看到每个商品的结算价和总价值
- 支付功能
- 订单功能

分解案例

■ 需求的分解

功能类型	需求功能	功能描述	详细描述
APP前端	首页	展示什么	如何展示
APP前端	商品列表	展示什么	如何展示
APP前端	订单列表	展示什么	如何展示
支付模块	微信支付	在哪儿支持微信支付	
支付模块	支付宝支付	在哪儿支持支付宝支付	
支付模块	银联支付	在哪儿支持银联支付	
后台	订单记录	记录什么	保存还是展示？ 保存在哪儿？以什么样的形式保存？展示在哪儿？怎么展示？

1.1. 分析模型类型：业务模型

■ 问题世界与业务模型

- 使用问题域中的**重要概念**作为模型的组元
- 使用概念之间的**业务联系**作为组元之间的关系

■ 使用了**业务描述**的方式，具有非形式化特征

- 可以抽取出需求信息中最重要和最本质的内容
- 可以达成用户和开发者的共同理解
- 业务模型元素（即业务概念和业务联系）的选取和定义上具有**不准确、不确定和模糊化**

■ 非形式化特征使得它不适合于进行需求建模

- **不足以描述**一个有效的软件解决方案
 - 不准确、不确定和模糊化

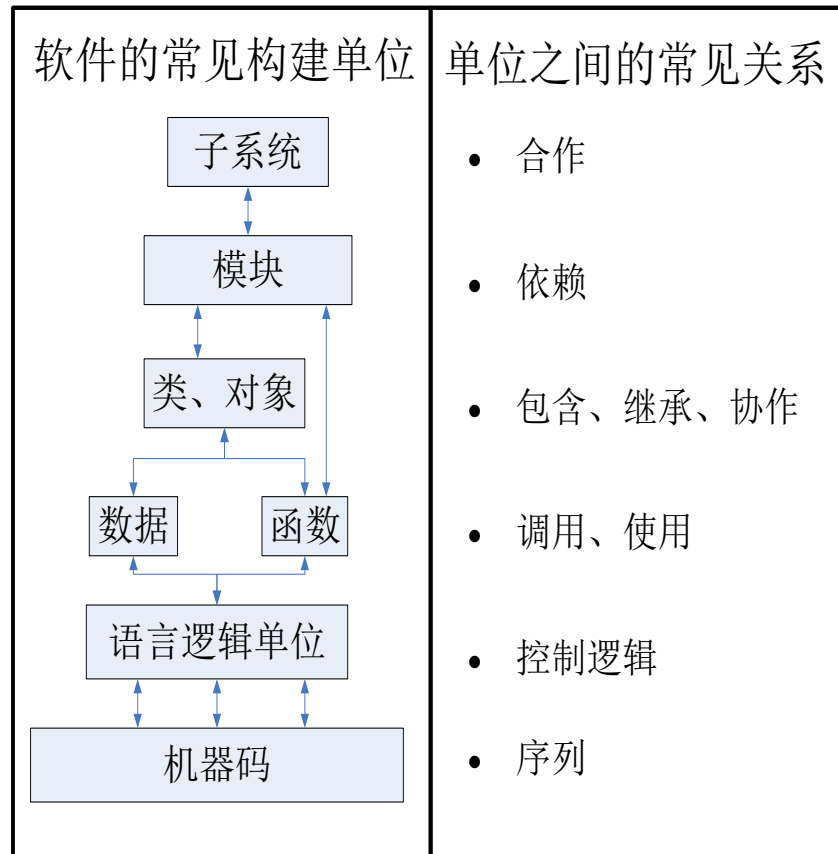
1.1.分析模型类型： 计算模型

■ 计算世界与计算模型

- 使用软件的构成单位作为模型的组元
- 软件构建单位之间的关系作为模型组元之间的关系

■ 基于计算科学建立的，具有形式化的特征

- 信息的描述具有明确化、准确化和确定化的特征



1.1.分析模型类型: 软件分析模型

■ 软件分析模型

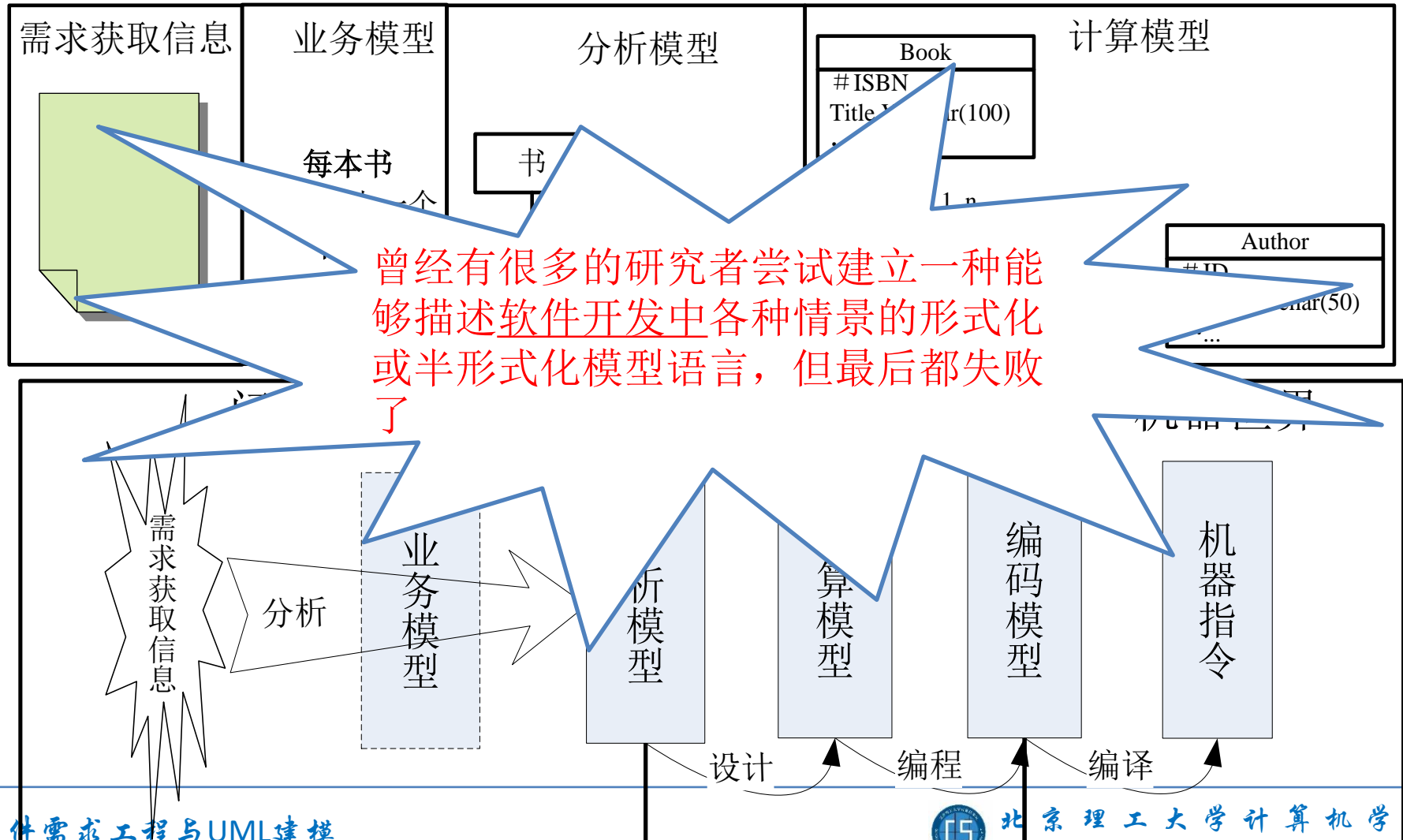
- 介于业务模型和计算模型二者之间的模型形式
- 使用了计算模型的组元形式
- 在组元的表现上采用了业务模型的表现方式

■ 半形式化的

- 不像计算模型那么严谨
- 比业务模型更严格

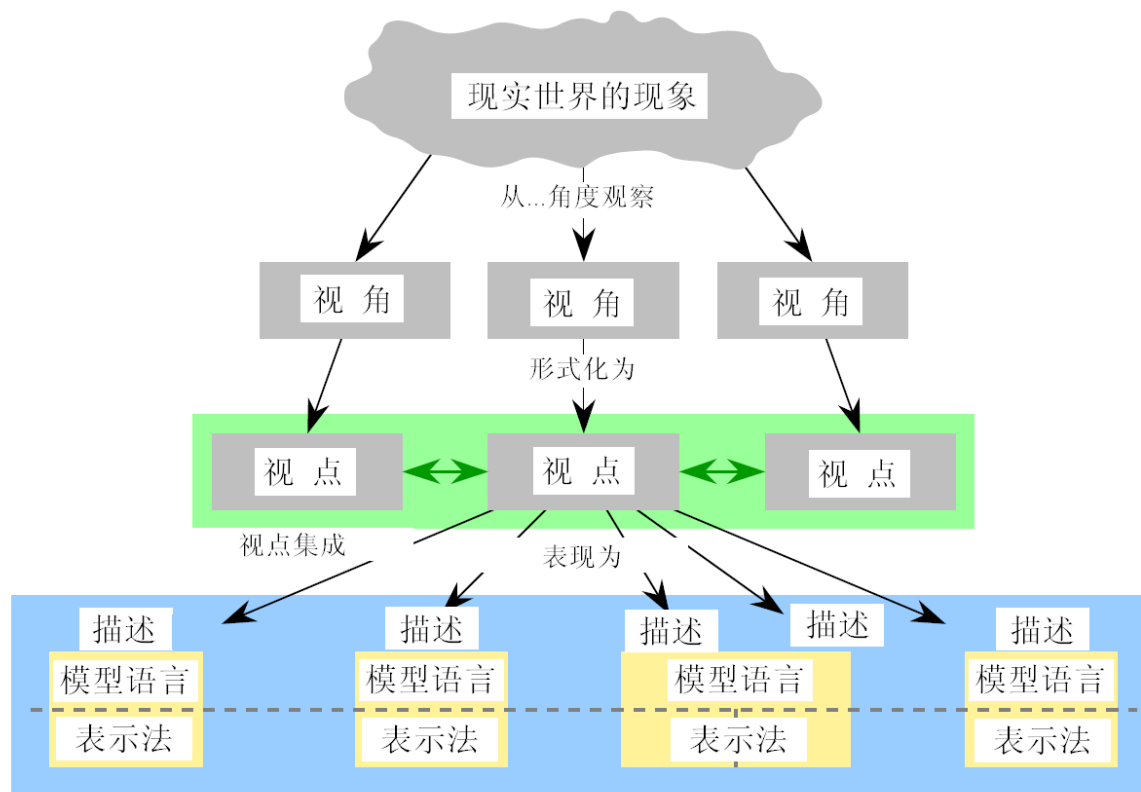
1.1. 建立分析模型

■ 三种模型



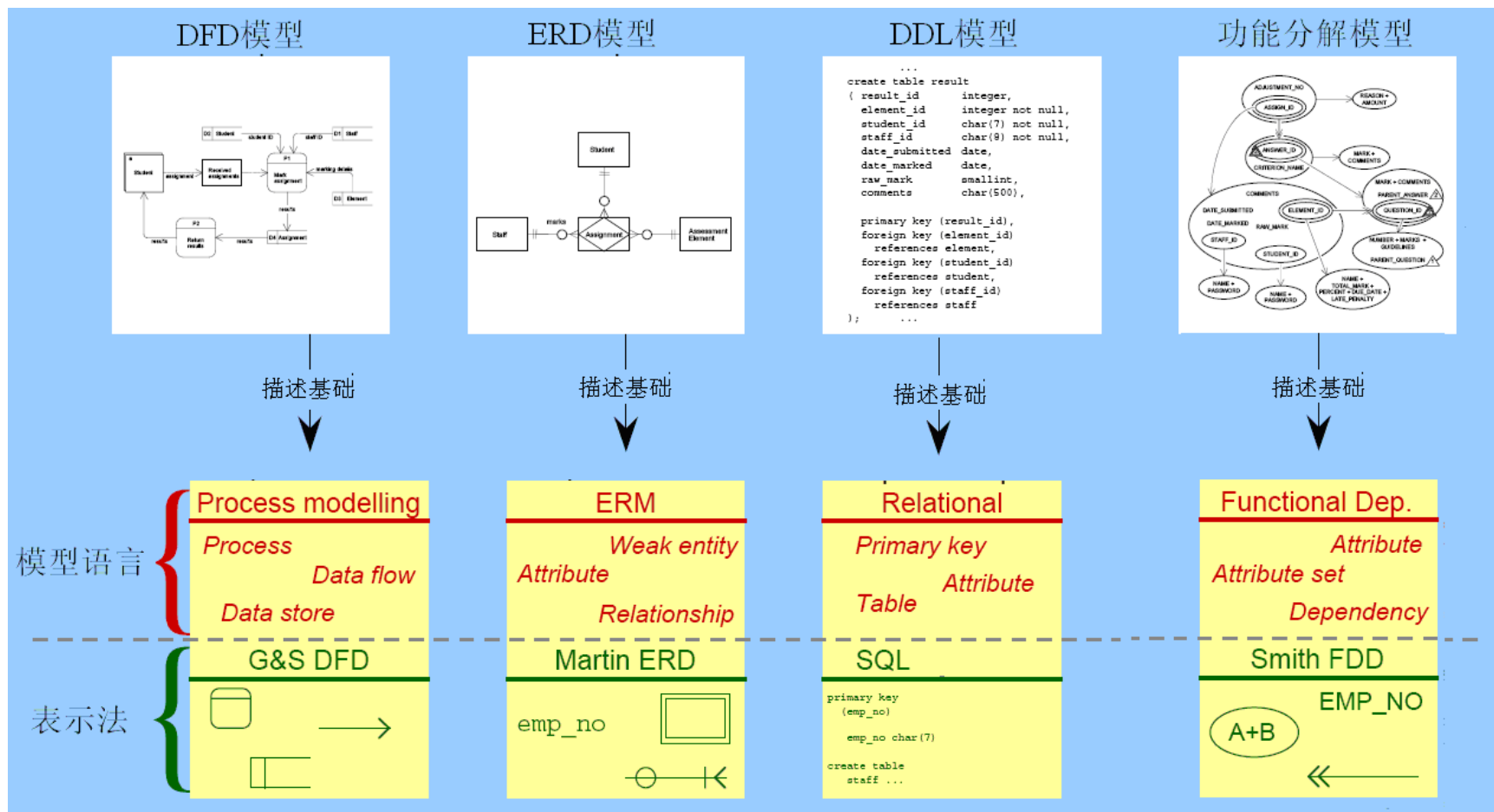
1.1. 建立软件分析模型：视点

- 模型的描述
 - 多视点方法

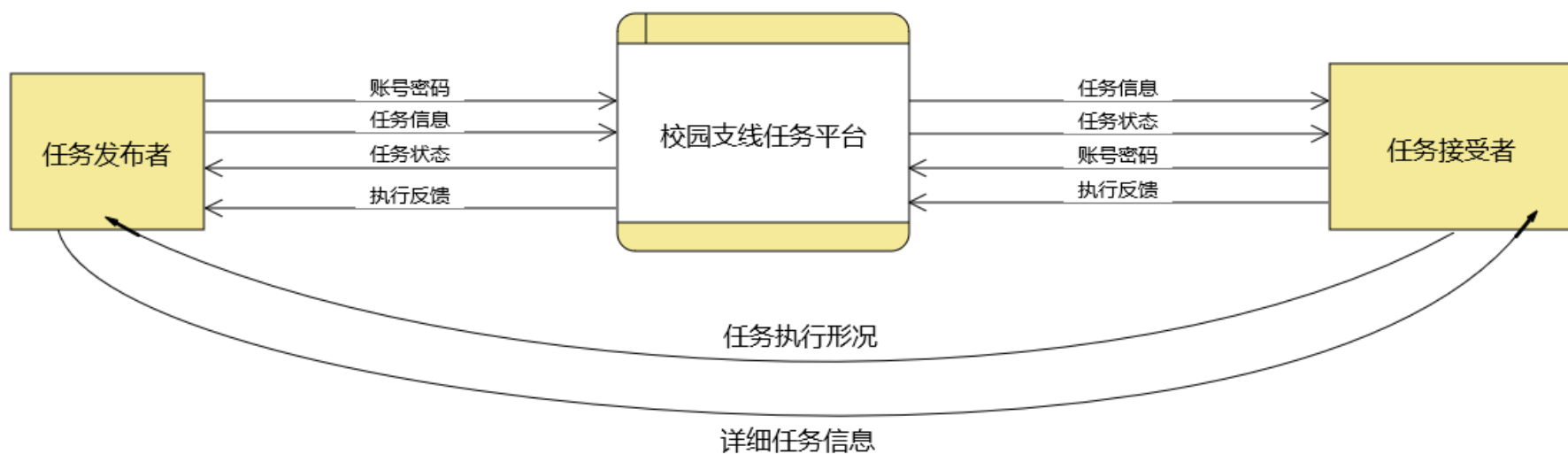


1.1. 建立软件分析模型

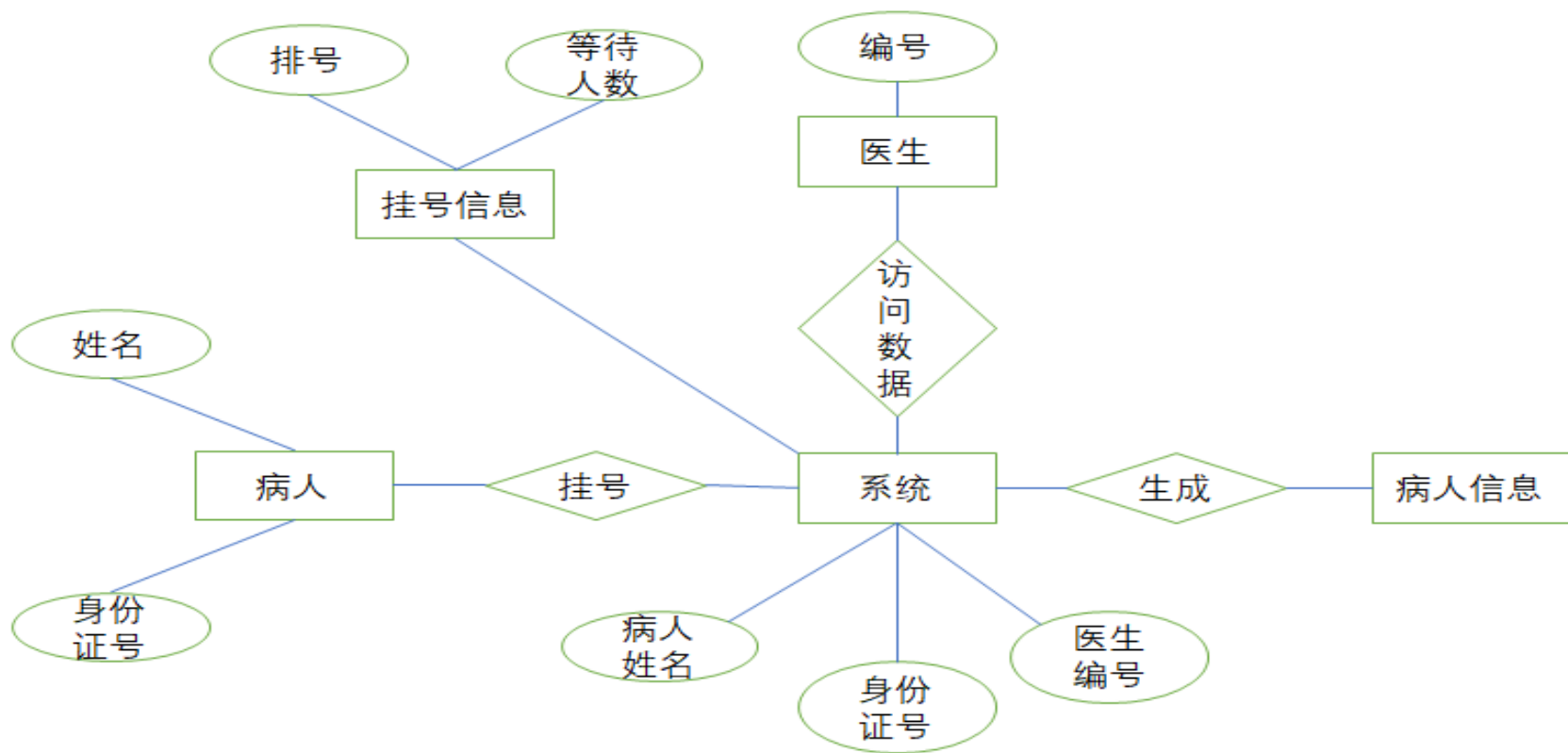
——模型、模型语言与表示法



校园支线任务平台-DFD模型



医院数据统计系统-ERD模型



DDL模型

■ 以网购平台为例

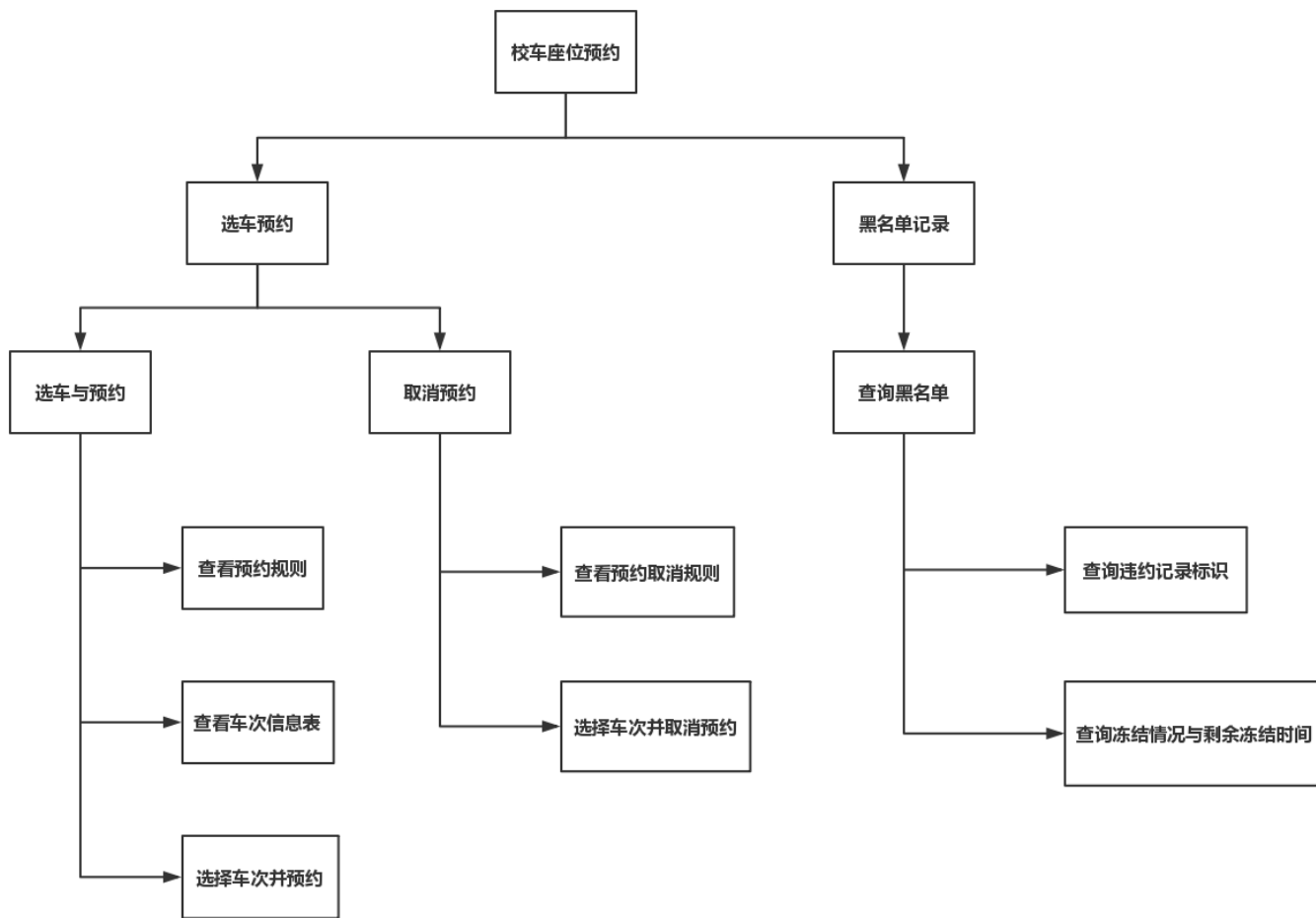
■ 创建用户数据表

```
CREATE TABLE Persons  
(  
  Id_P INT NOT NULL UNIQUE,  
  LastName VARCHAR(255) NOT NULL,  
  FirstName VARCHAR(255),  
  Address VARCHAR(255),  
  City VARCHAR(255)  
);
```

■ 订单数据表

```
CREATE TABLE Orders(  
  O_Id INT NOT NULL PRIMARY KEY,  
  OrderNo INT NOT NULL,  
  Id_P INT  
);
```

校车座位预约系统-功能分解模型

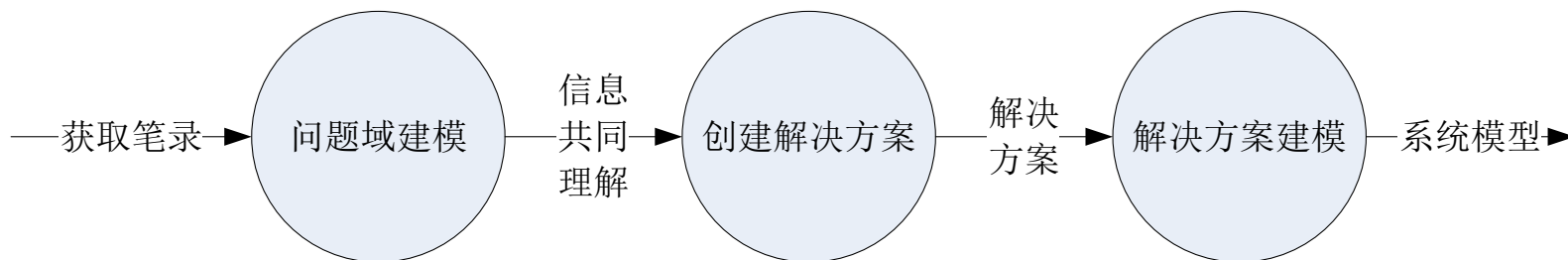


1.1. 建立分析模型过程

■ 需求建模

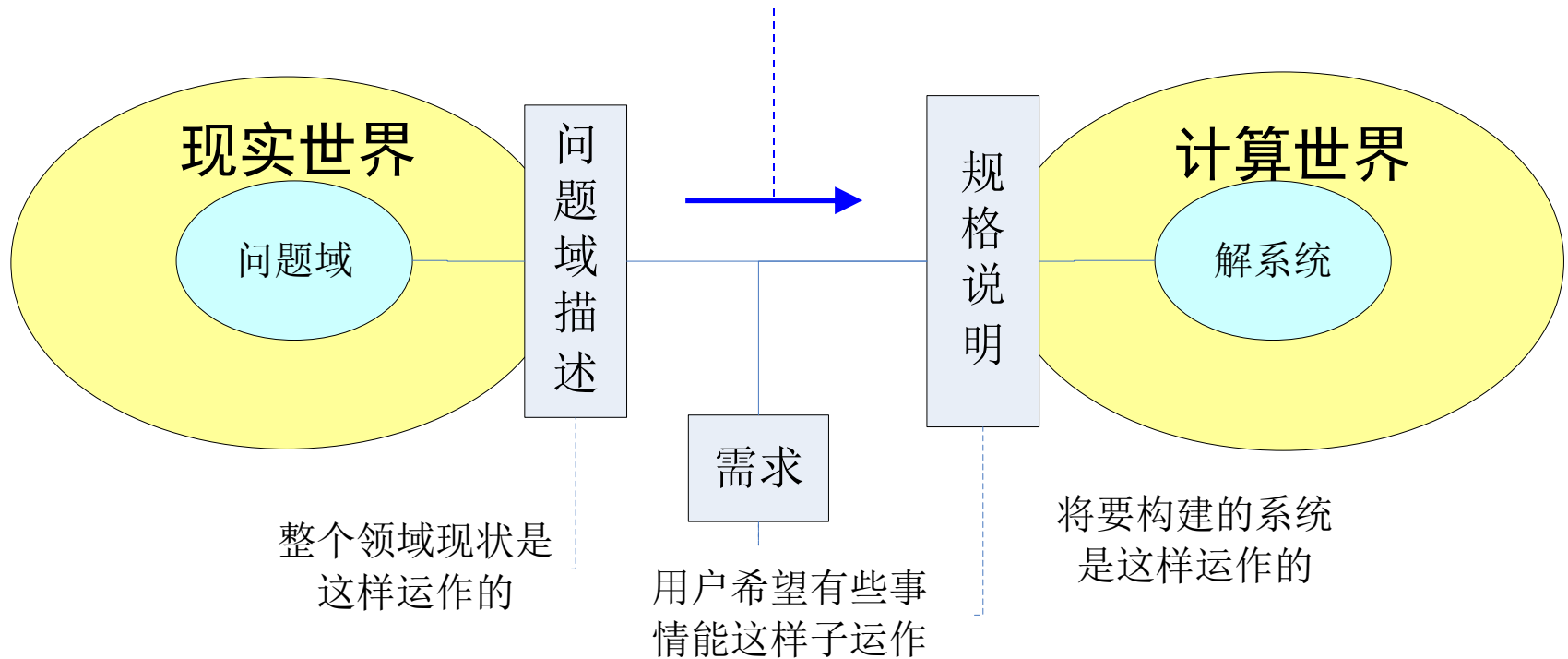
■ 通常的做法是：

- 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。

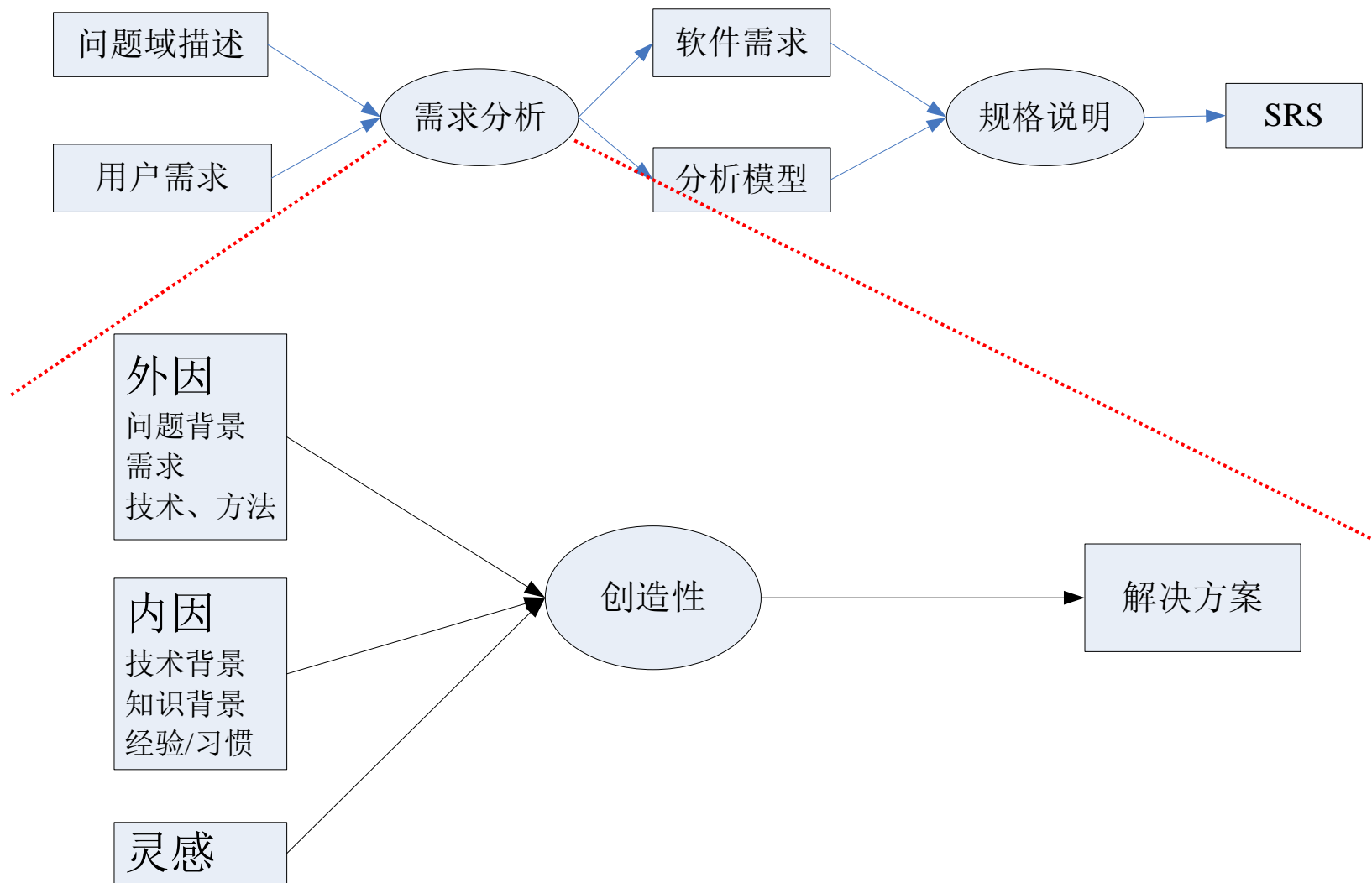


1.2. 建立解决方案

需求分析的目标



1.2 建立解决方案的过程



主要内容

1. 需求分析的根本任务
2. 需求分析技术
 1. 常用需求分析技术
 2. 需求分析技术的发展过程
3. 需求分析方法
4. 前期需求阶段的建模与分析
5. 需求分析的活动

2.1. 常用需求分析技术

■ 结构化技术

■ 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

■ 数据建模

- 实体关系图Entity Relationship Diagram

■ 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

2.1 常用需求分析技术

■ 技术的综合运用

■ 如何为各个视角选择需求分析技术？

- 每一种需求分析技术都有自己的特点，具有在应用上的独特性

■ 如何实现它们之间的配合？

- 只有通过多种需求分析技术的有机结合与集成才能充分的描述复杂应用

2.2. 需求分析技术的发展过程

■ 传统分析

- 没有方法 (1950's)
 - 依赖个体才智,依据个人习惯
 - 缺乏结构、不可重复、不可测量,冗长、混乱、偏颇、无结构等等

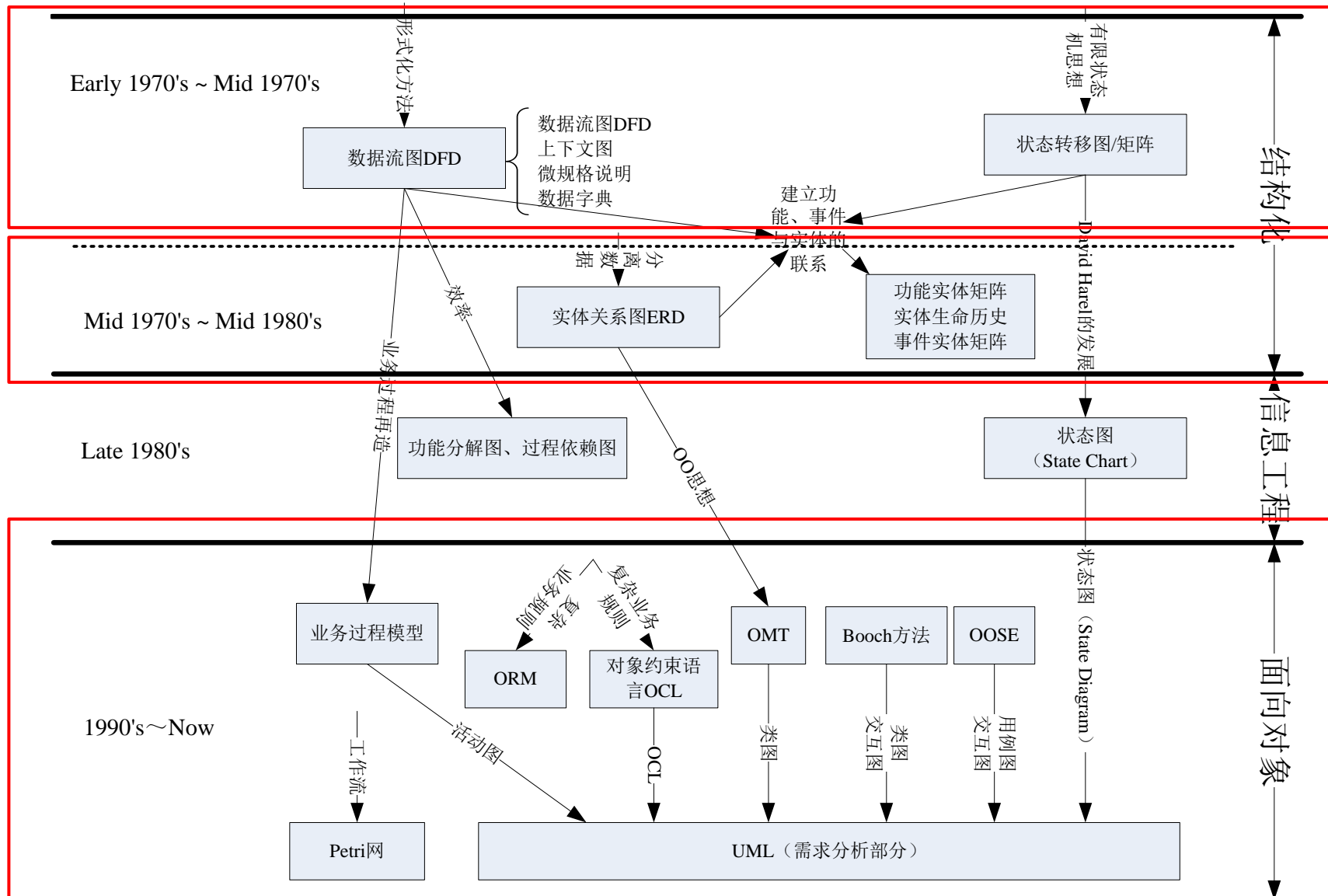
■ 结构化分析

- 传统结构化分析 (late 1960's),现代结构化分析 (late 1970's)
 - 以数据流动为中心,以DFD为核心技术,辅助ERD,STD...
- 信息工程 (late 1980's)
 - 以数据知识结构为基础,ERD为核心技术,辅助DFD, STD, FDD, PD...

■ 面向对象分析 (1990's)

- 以对象为中心,以UML(类图)为核心技术
- 以全面思想革新为理想,以承继结构化技术为现实

2.2. 需求分析技术的发展过程

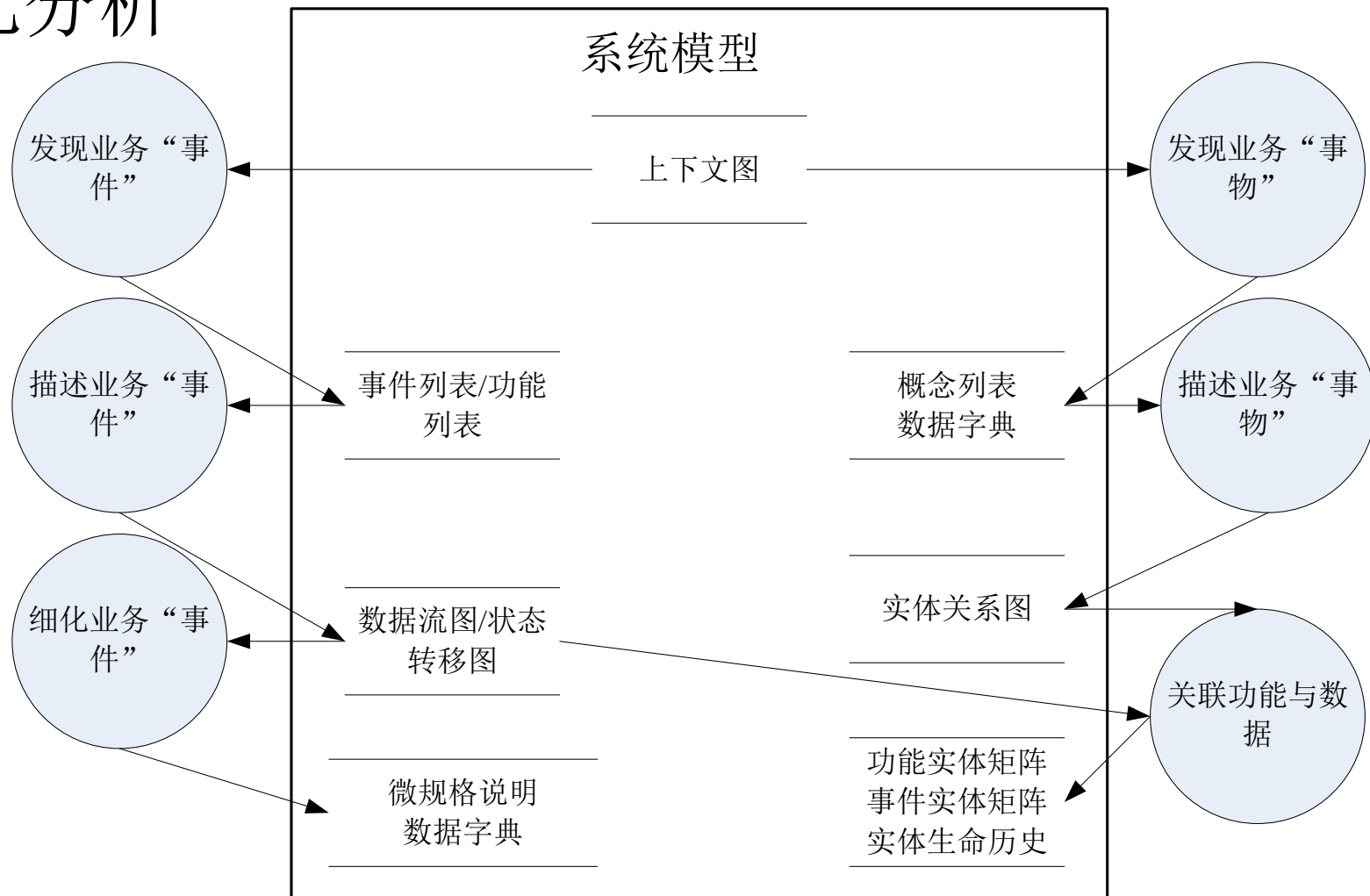


主要内容

1. 需求分析的根本任务
2. 需求分析技术
3. 需求分析方法
4. 前期需求阶段的建模与分析
5. 需求分析的活动

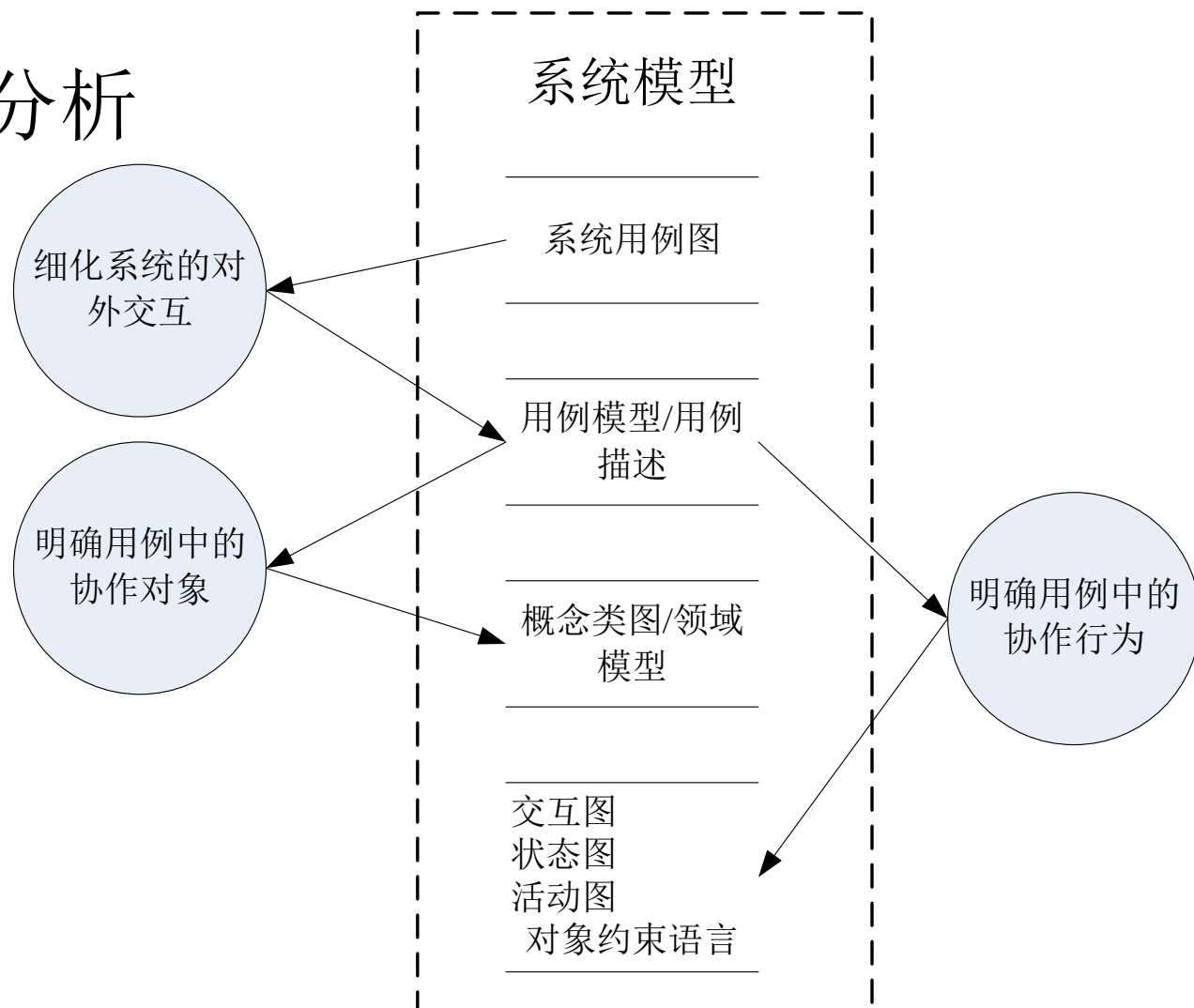
3. 需求分析方法

■ 结构化分析



3. 需求分析方法

■ 面向对象分析

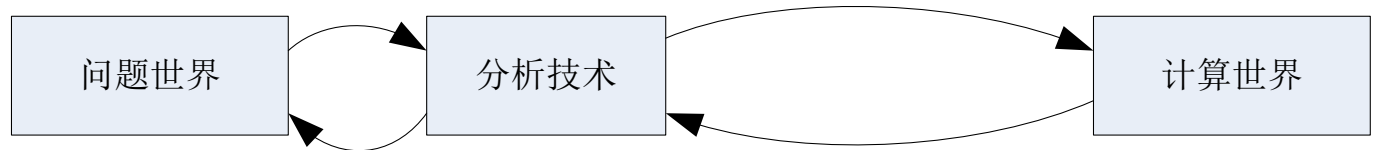


主要内容

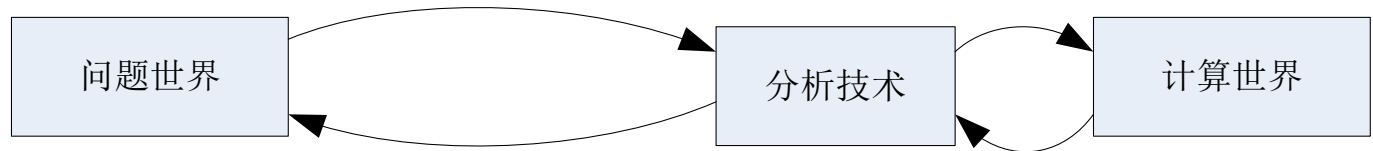
1. 需求分析的根本任务
2. 需求分析技术
3. 需求分析方法
4. 前期需求阶段的建模与分析
5. 需求分析的活动

面向问题与面向解系统

面向问题

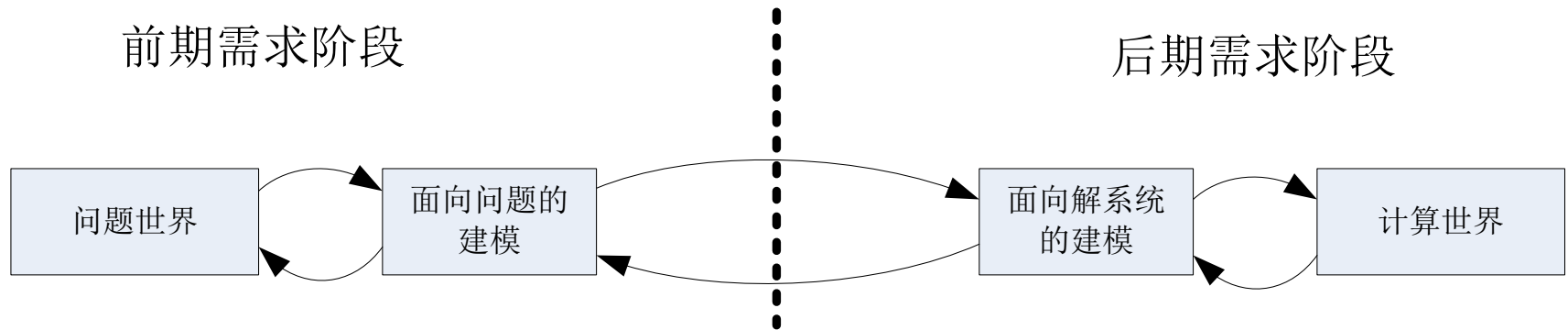


面向解系统



前期需求阶段

后期需求阶段



面向问题域的分析

■ 面向问题域的分析

■ 问题框架

- 特性
- 解决

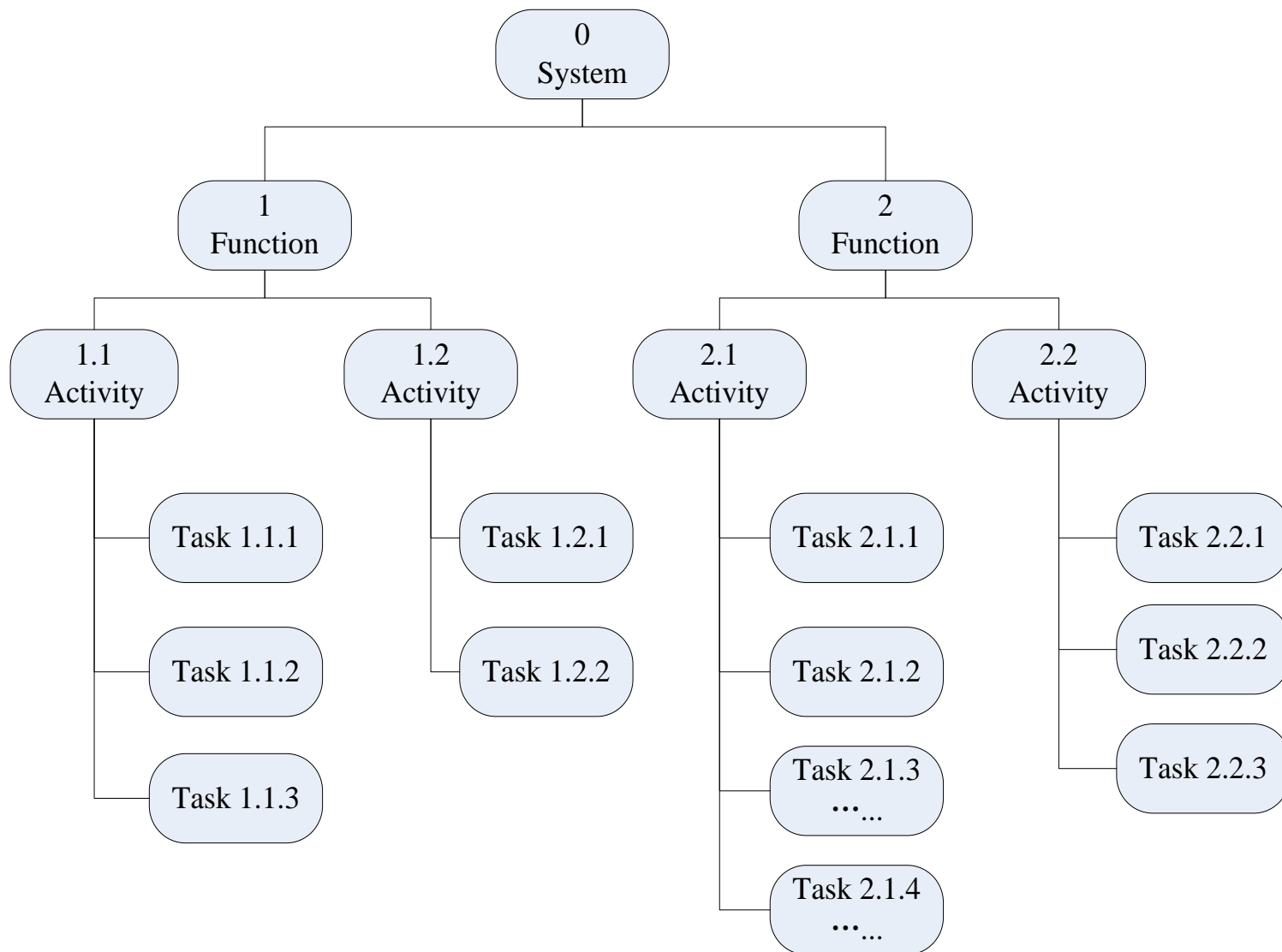
■ 基本思路

- 研究所有可能的问题域，从中发现一些重复出现的简单问题类型
- 分析每一种问题框架的特性，确定问题的理解和解决方法
- 将问题框架的建立和分类系统化，以简单的问题框架为基本单位，进行复杂问题的分解

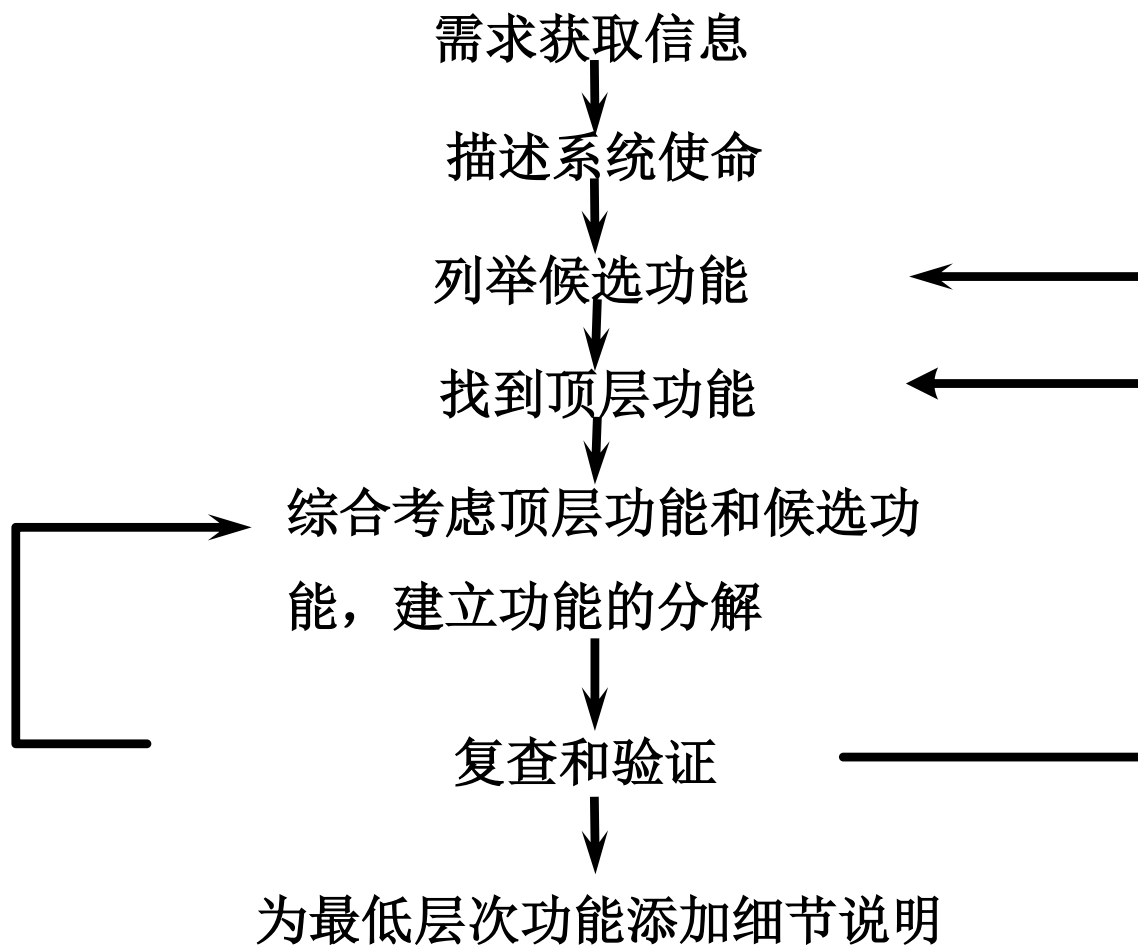
功能分解图

- 在一个图内自上至下的集中显示系统的功能分解结构
 - 最顶层的单独功能通常是对整个系统的使命描述，是对系统业务需求的概括
 - 系统使命说明的下一层被称为功能的最顶层，描述了系统应该具备的一些重要功能，它们支撑着系统使命的实现
 - 功能最顶层下面的分支是对最顶层功能执行分解后形成的层次关系
 - 最底层的是基本的业务功能。这些基本的业务功能是人们所能找到的最基本的、不可再细分的功能或处理
- 能够更加集中、更加直观的展示大量过程之间的层次关系

功能分解图



功能分解过程



案例1-学习资料分享平台

■ 系统使命

- 学生之间进行资料共享的平台

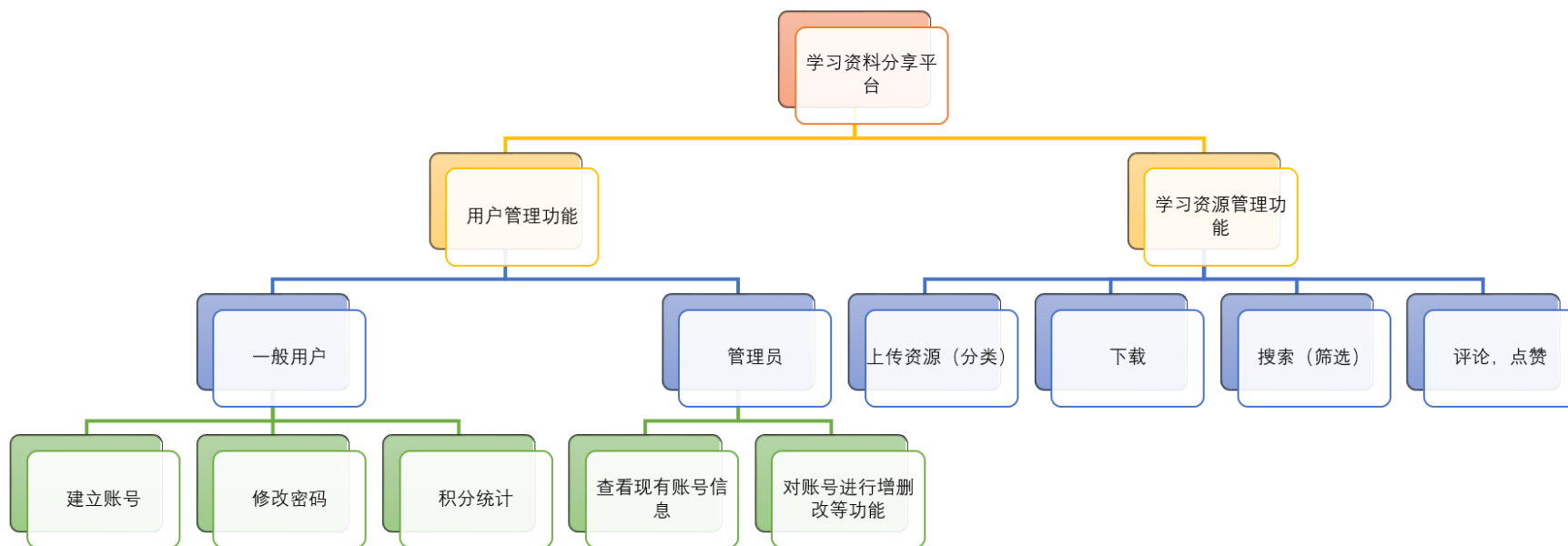
■ 功能的最顶层

- 用户管理
- 学习资料管理
- ...

■ 基本业务功能

- 资源上传、下载
- 资源评分
- 资源搜索
- 评论、点赞功能
- ...

案例1-学习资料分享平台



案例2-校园答疑系统

■ 系统使命

- 为老师与学生提供线上知识交流的平台（除了专业知识外，也提供对社会热点问题的评论交流）

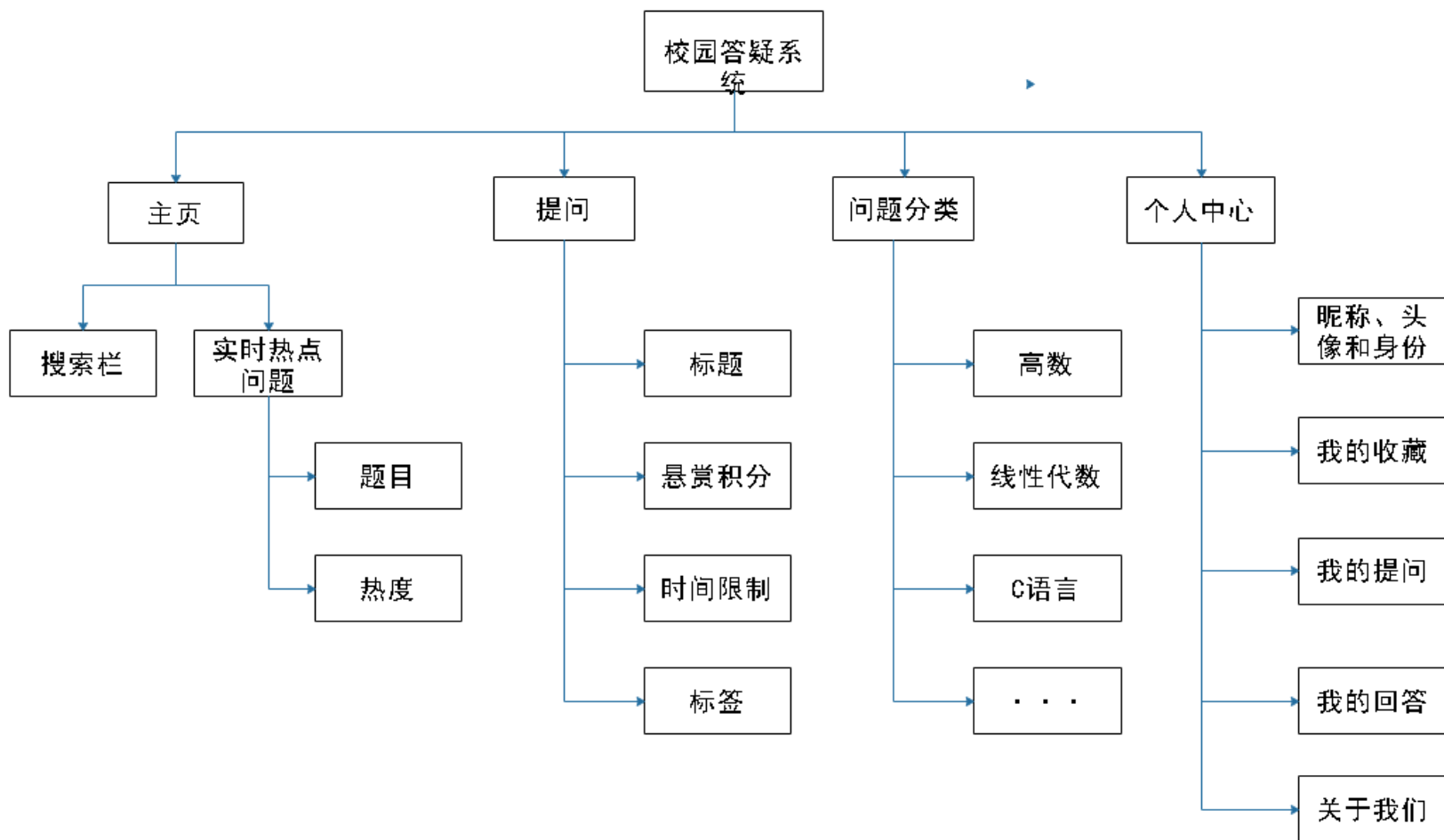
■ 功能的最顶层

- 系统主页展示
- 提问功能
- 问题分类
- 用户个人中心管理功能

■ 基本业务功能

- 问题搜索功能
- 热点问题实时排行
- 悬赏积分制
- 对问题进行时间限制
- 查看本人提出的问题
- 查看本人作出的回答
- 查看收藏的问题
- ...

案例2-校园答疑系统



案例3-时间管理助手

■ 系统使命

- 帮助用户提高对任务和时间管理能力

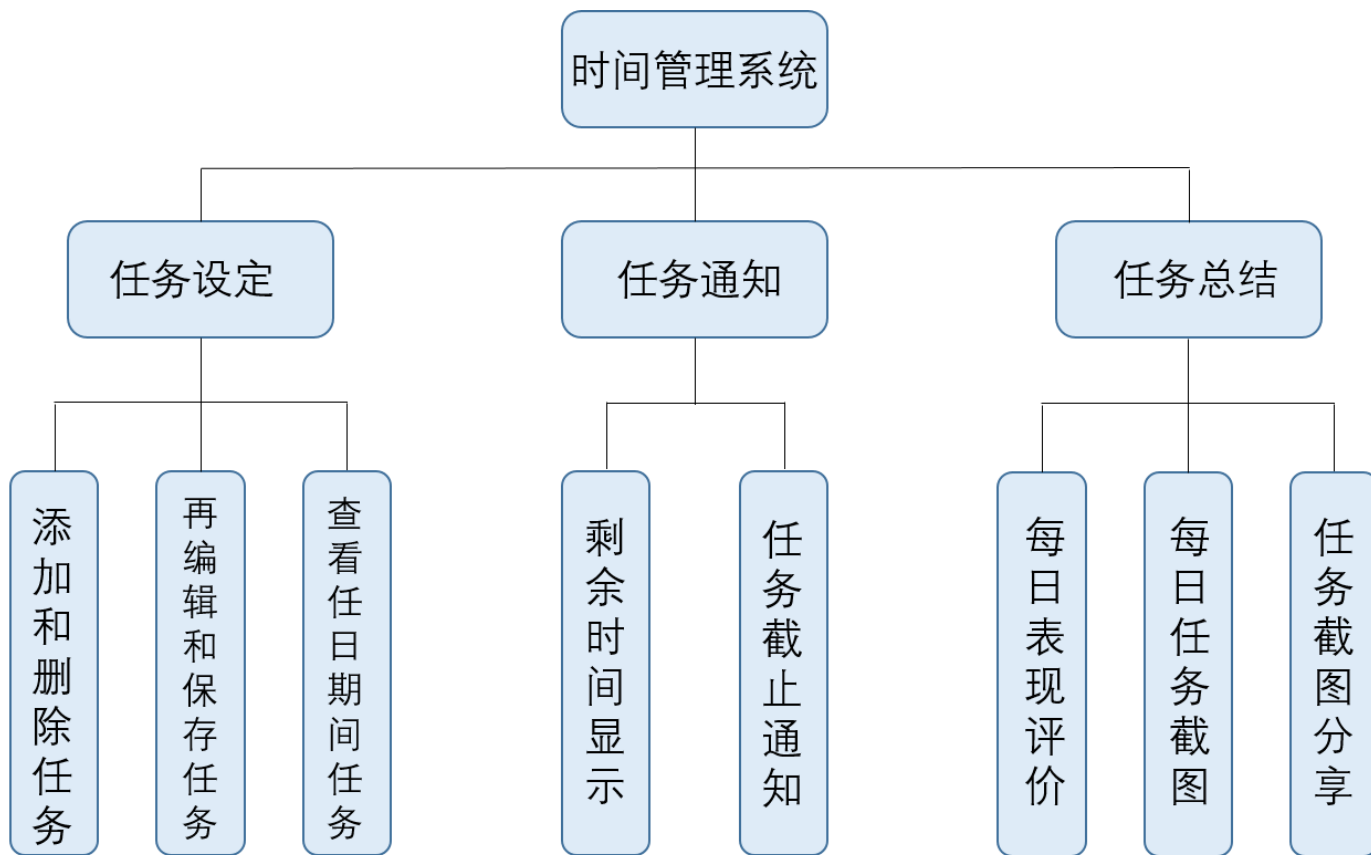
■ 功能的最顶层

- 任务设定
- 任务通知
- 任务总结

■ 基本业务功能

- 任务的增删改
- 任务剩余时间提醒
- 任务截止通知
- 每日表现评价功能
- 任务截图分享功能
- ...

案例3-时间管理助手



案例4-失物招领系统

■ 系统使命

- 为同学们提供发布和寻找丢失物品的平台

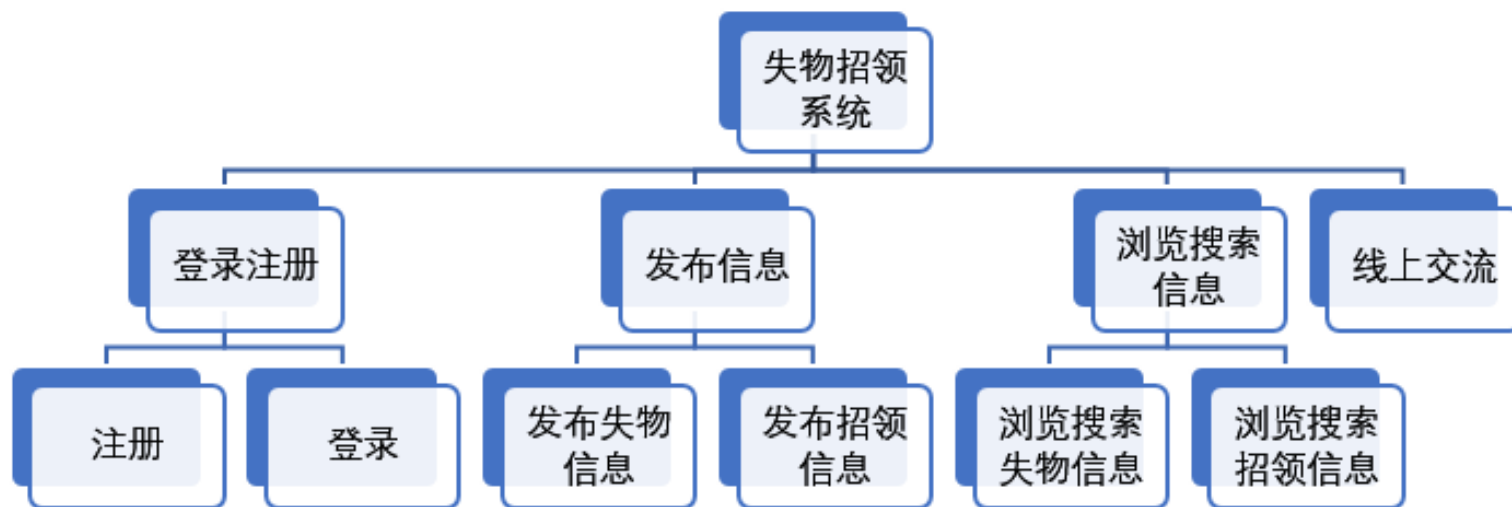
■ 功能的最顶层

- 用户登录、注册
- 发布失物信息
- 浏览失物信息
- 用户之间的线上交流（用以确认失物）

■ 基本业务功能

- 注册
- 登录
- 发布失物信息
- 发布招领信息
- 浏览失物信息
- 浏览招领信息
- ...

案例4-失物招领系统



案例5-校内二手交易平台

■ 系统使命

- 为同学们提供二手商品交易的平台

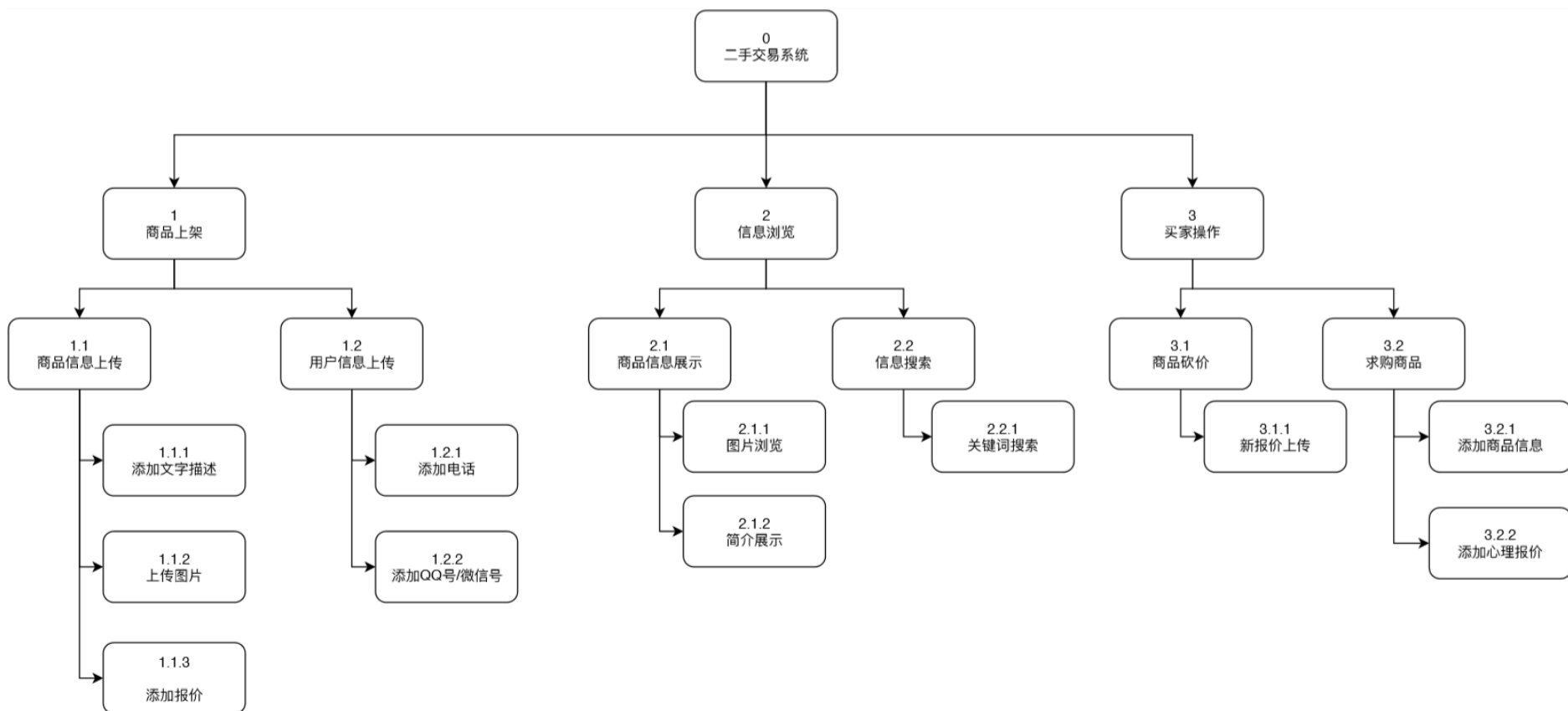
■ 功能的最顶层

- 商品上架（卖家操作）
- 信息浏览
- 买家操作

■ 基本业务功能

- 商品信息上传（图片、文字描述、报价等）
- 用户信息上传（用以买卖双方直接沟通）
- 关键词搜索功能
- 商品砍价功能
- 商品求购功能
- ...

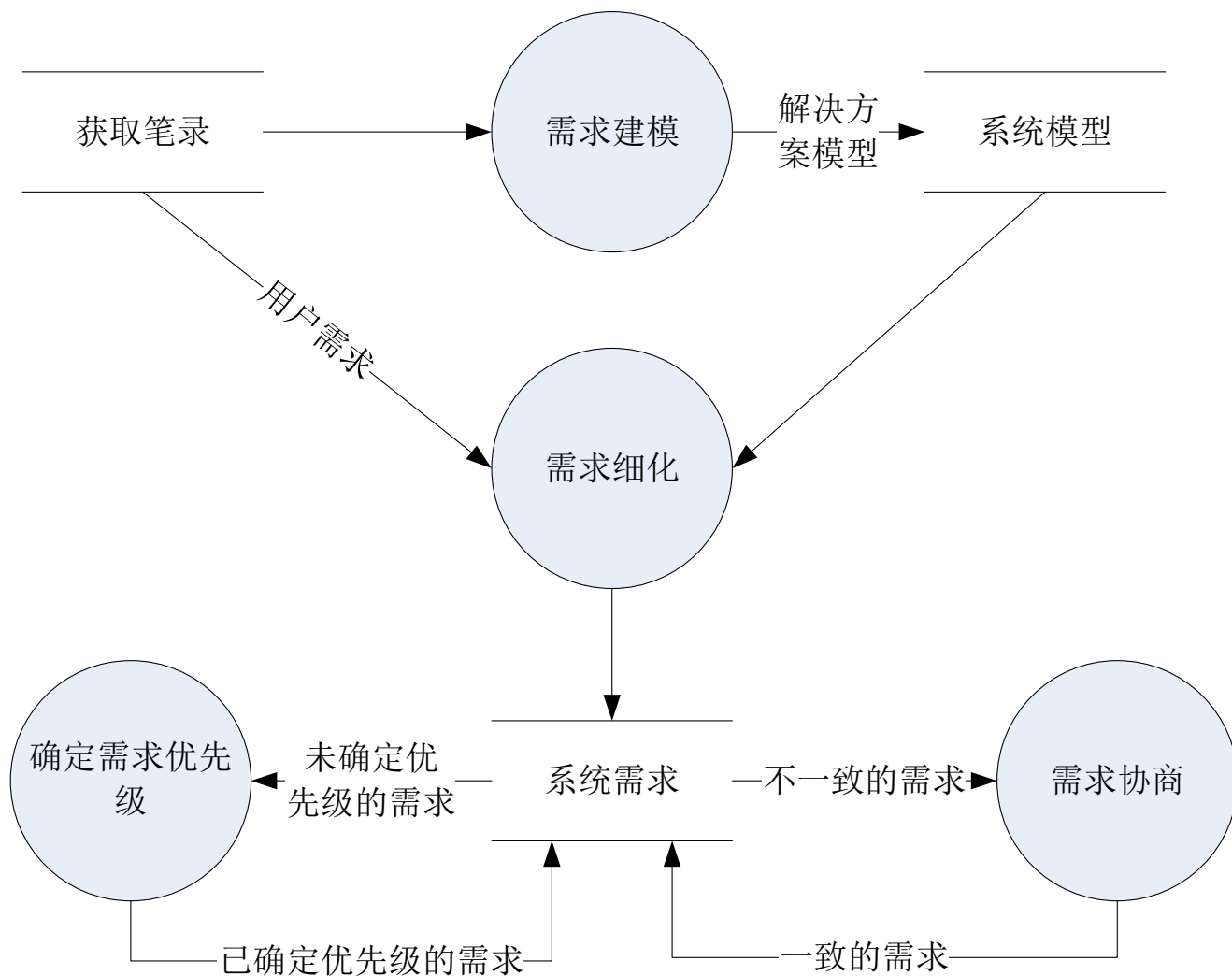
案例5-校内二手交易平台



主要内容

1. 需求分析的根本任务
2. 需求分析技术
3. 需求分析方法
4. 前期需求阶段的建模与分析
5. 需求分析的活动

5. 需求分析的活动



5. 需求分析的活动 ——需求细化

- 明确用户需求的隐含因素
- 将从问题域和业务的角度表述的用户需求等价的转化为从软件和技术角度表述的系统需求
- 非功能需求也需要从高层次的表述方式转化为一系列更加详细和具体的需求表述
- 需求细化也会发现新的细节需求
- 细化后的需求应该被一一的标识和记录下来

5. 需求分析的活动 ——需求细化

■ 需求的记录

- 标识符（ID），每一条需求都应该能够通过ID唯一的标识自己。
- 源头（Source），要能够回溯到需求的源头，例如特定的涉众。
- 理由（Rational），需求被提出的目的。
- 优先级（Priority），详细情况见下一节。
- 成本（Cost），预估的实现成本。
- 风险（Risk），实现该需求的过程中可能带来的风险。
- 可变性（Volatility），将来发生变化的可能性。

需求细化案例-记账本APP

- 除了基本的电子记账功能外甲方提出的其他需求
 - 实现账单可视化统计分析功能
 - 人机交互功能
 - 账目情景设置可以多样化（例如吃饭、购物、游乐场等）
 - 删除记错的账单

需求编号	A1
需求描述	记账界面可人机交流
源头	针对甲方对于记账可交流的要求
成本需要	在现有记账页面的基础上，增加一个记账成功的回复功能
可变性	高
优先级	高

需求编号	B1
需求描述	消费趋势可视化
源头	针对用户对于消费了解到需求
成本需要	对记录的数据进行统计分析，并可视化
可变性	较高
优先级	较高

需求编号	A3
需求描述	对记错的帐可以进行删除
源头	针对用户对于记账的需求要求
成本需要	对已完成的记账添加删除操作
可变性	中等
优先级	低

需求编号	C1
需求描述	情景设置可以多样化
源头	甲方对于情景的多样化需求
成本需要	在已有的情景中添加多样的选择
可变性	较高
优先级	中等

需求优先级案例-记账本APP

■ 需求优先级排名-TOP4

优先级	需求任务
第一优先级	A1记账界面可人机交流
第二优先级	B1消费趋势可视化
第三优先级	C1情景设置可以多样化
第四优先级	A3对记错的帐可以删除

5. 需求分析的活动

——确定需求优先级

■ 累计投票

■ 区域划分

- 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。

	重要	不重要
紧急	高优先级	不予处理
不紧急	中优先级	低优先级

5. 确定需求优先级案例

■ 数据量化

$$\text{优先级} = \frac{\text{价值}\%}{(\text{成本}\% \times \text{成本权值}) + (\text{风险}\% \times \text{风险权值})}$$

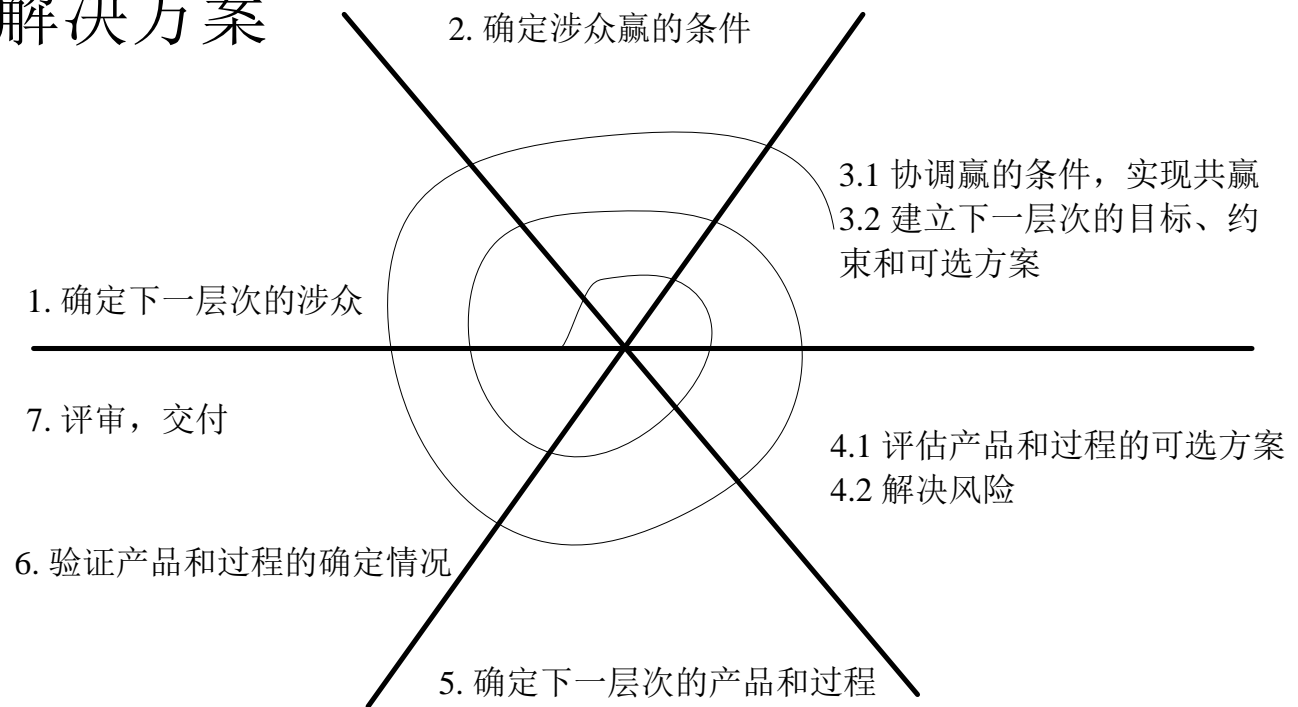
相对权值	2	1			1		0.5		
需求	相对收益	相对损失	总价值	价值%	相对成本	成本%	相对风险	风险%	优先级
1.打印化学品安全数据表格	2	4	8	5.2	1	2.7	1	3.0	1.22
2. 查询供应商的订单状态	5	3	13	8.4	2	5.4	1	3.0	1.21
3. 生成化学品仓库存货清单报表	9	7	25	16.1	5	13.5	3	9.1	0.89
4. 查看某个特定化学品容器的历史记录	5	5	15	9.7	3	8.1	2	6.1	0.87
.....
总计	53	49	155	100	37	100	33	100	

需求优先级案例-校车座位预约系统

相对权值	2	1			1		0.5		
需求	相对收益	相对损失	总价值	价值%	相对成本	成本%	相对风险	风险%	优先级
1、获取校车班次信息表	5	3	13	12.3	2	13.3	1	8.3	0.70
2、实名制认证注册窗口	4	2	10	9.4	1	6.7	2	16.7	0.66
3、个人黑名单记录页	5	4	14	13.2	2	13.3	2	16.7	0.61
4、选车次预约	9	6	24	22.6	3	20	1	8.3	0.94
5、选车次取消预约	7	3	17	16	3	20	1	8.3	0.66
6、意见反馈窗口	2	4	8	7.5	1	6.7	2	16.7	0.50
7、账户冻结情况查询	4	4	12	11.3	2	13.3	2	16.7	0.52
8、查看预约与取消预约的规则	3	2	8	7.5	1	6.7	1	8.3	0.69
总计	39	28	106	100	15	100	12	100	

5. 需求分析的活动 ——需求协商

- 明确冲突的因素，避免情绪上的冲突
- 明确冲突的解决空间
- 确定最佳解决方案



本章小结

- 需求分析是需求工程中最重要和核心的活动，它对信息的建模是理解问题的关键，也是创建正确解决方案的关键
- 需求分析涉及很多的技术和方法，需求分析活动的有效执行需要分析人员能够掌握并判定这些方法的选择与使用
- 需求分析过程当中会执行很多的重要子活动，它们的有效整合确保了整个需求分析工作的成功

思考题1

- 需求分析的根本任务是什么？需求分析阶段需要执行哪些活动？

思考题1

- 需求分析的根本任务是什么？需求分析阶段需要执行哪些活动？
 - 需求分析的根本任务：1. 建立分析模型 2. 建立解决方案
 - 需要执行的活动：
 - 问题分析
 - 确定需求边界
 - 需求建模
 - 需求细化
 - 确定需求优先级
 - 需求协商

思考题2

- 简述一下需求分析的原则？

思考题2

- 简述一下需求分析的原则？
 - 必须能够表达和理解问题的数据域和功能域
 - 按自顶向下、逐层分解问题
 - 要给出系统的逻辑视图和物理视图

思考题3

- “事件”和“事物”一直是进行需求分析的一个重要思路，你对此如何评价？

思考题3

- “事件”和“事物”一直是进行需求分析的一个重要思路，你对此如何评价？
 - **事件：**可以描述、值得记录的在某一个特定时间和地点发生的事情。通过对事件的分析可以将复杂的系统需求分解成易处理并能更好理解的小单元。事件可以分为以下几类：外部事件、临时事件、状态事件。
 - **事物：**在面向对象的编程中，这些事物是在系统中相互交互的对象。事物的类型：实物、角色、组织部门、设备、突发事件、事件或交互、地点、位置。

思考题3

- “事件”和“事物”一直是进行需求分析的一个重要思路，你对此如何评价？
 - 结构化分析：事物和事件正好对应着结构化方法的两条路径，一个是DFD，一个是ERD。
 - 面向对象分析：核心是多对象的协同，而多对象既不是事件也不是事物，而是基于场景的。