软件需求工程

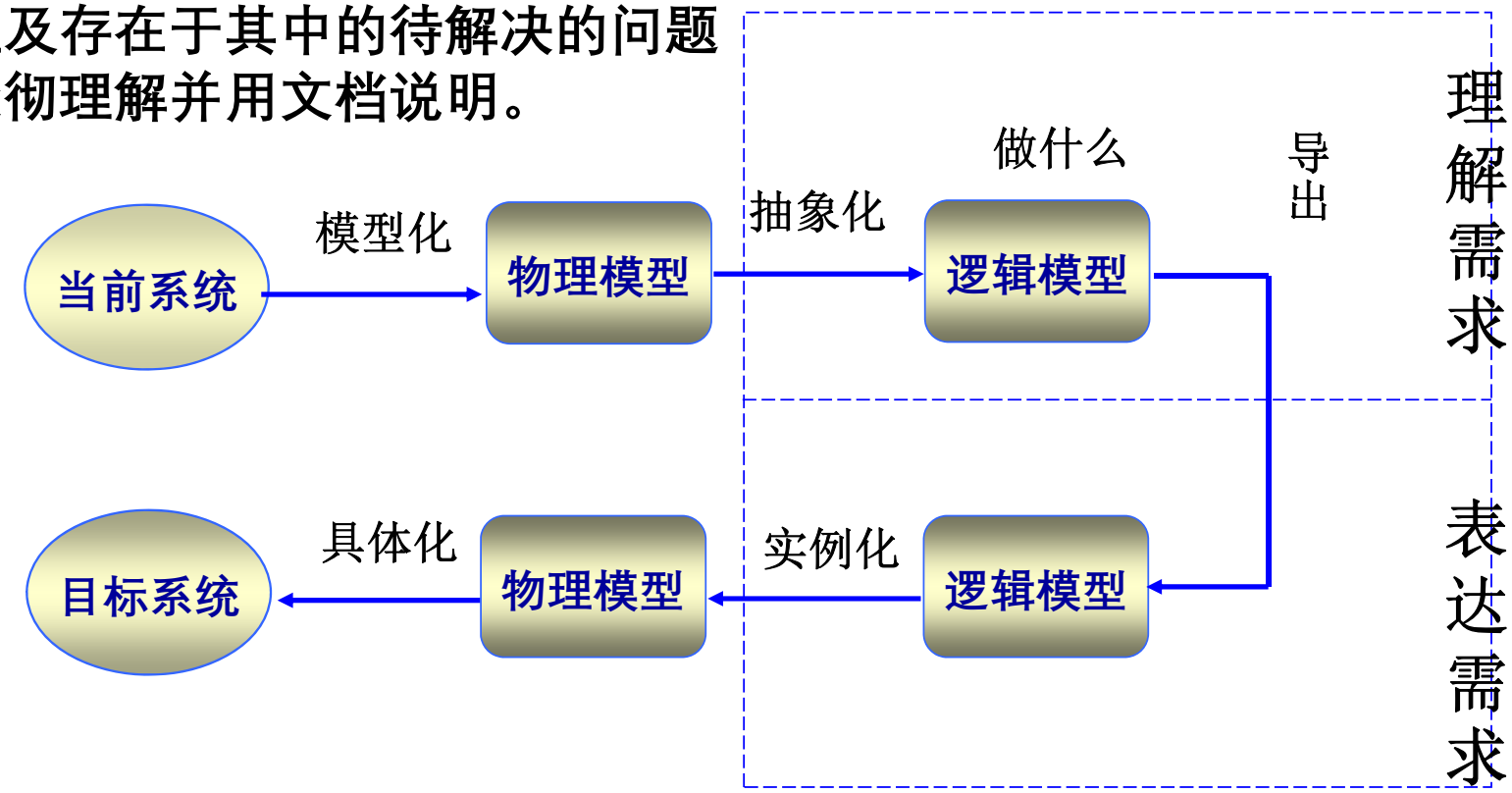
需求分析

需求分析的任务

- 简单地说，需求分析的基本任务就是分析和综合已收集到的需求信息。
- 分析的工作在于透过现象看本质，找出需求信息间的内在联系和可能的矛盾。
- 综合的工作在于去掉非本质的信息，找出解决矛盾的方法并建立系统的逻辑模型。

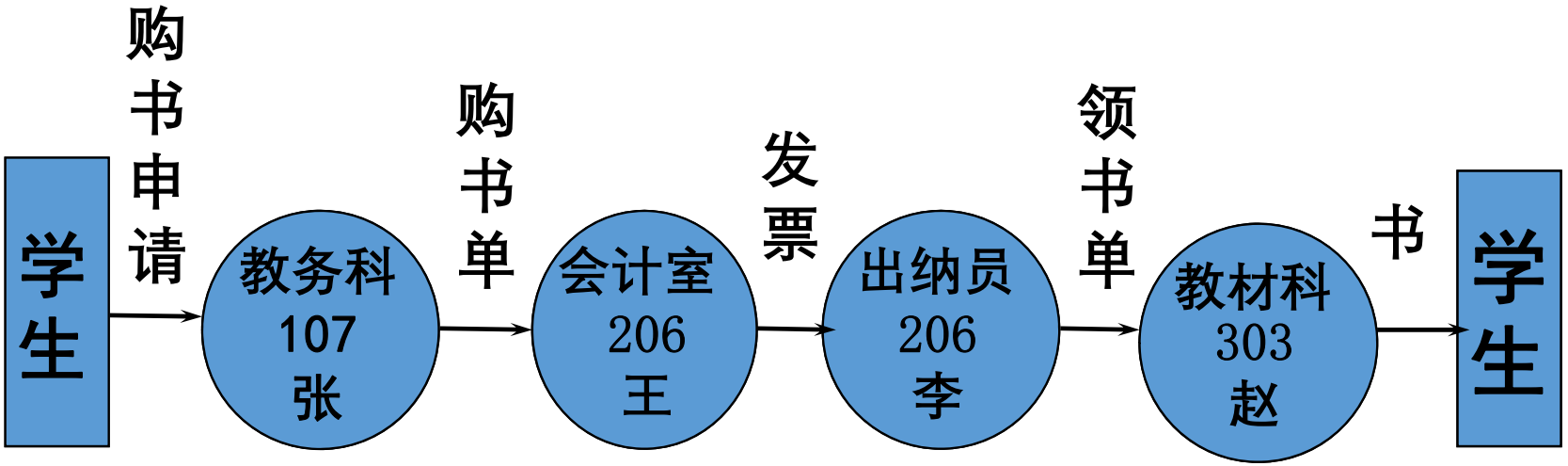
需求分析：从问题域到解决域

- **分析是指通过对问题域的研究，获得对该领域特性及存在于其中的待解决的问题特性的透彻理解并用文档说明。**



需求分析过程(1)

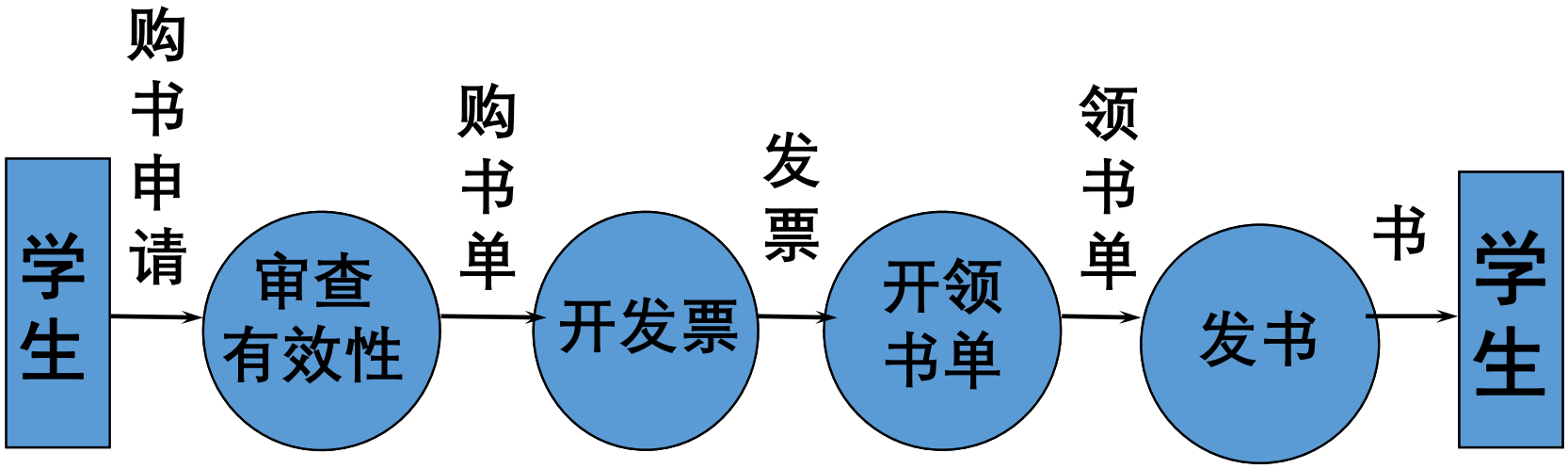
- 通过对现实环境的调查，获得当前系统的物理模型



学生购买教材的物理模型

需求分析过程(2)

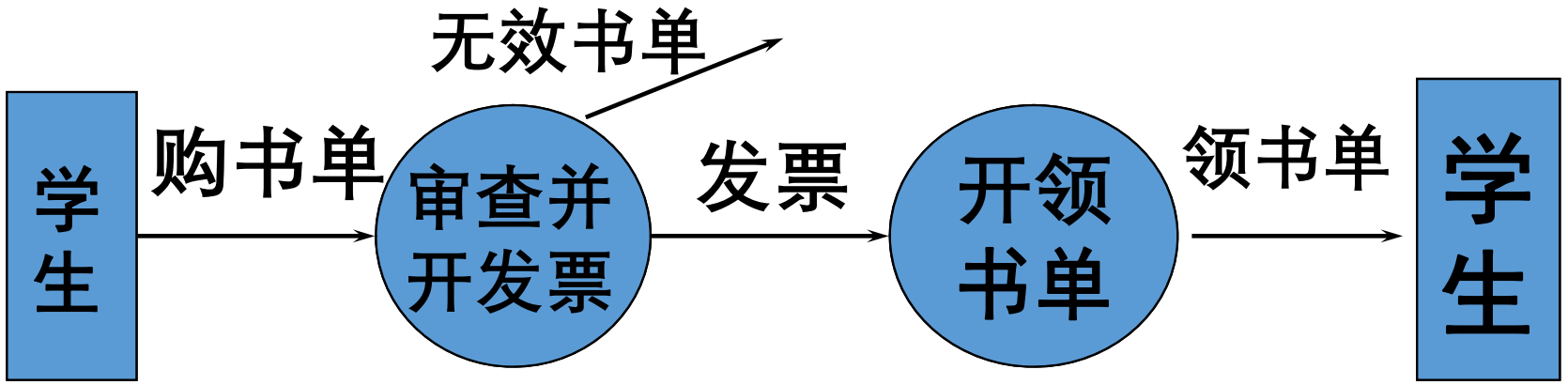
- 去掉具体模型中的非本质因素，抽象出当前系统的逻辑模型



学生购买教材的逻辑模型

需求分析过程(3)

- 分析当前系统与目标系统的差别，建立目标系统的逻辑模型



计算机售书系统的逻辑模型

软件需求分析

- 具体的说，需求分析的基本任务是提炼、分析和仔细审查已收集到的需求信息，找出真正的、具体的需求，以确保所有项目相关人员都明白其含义。

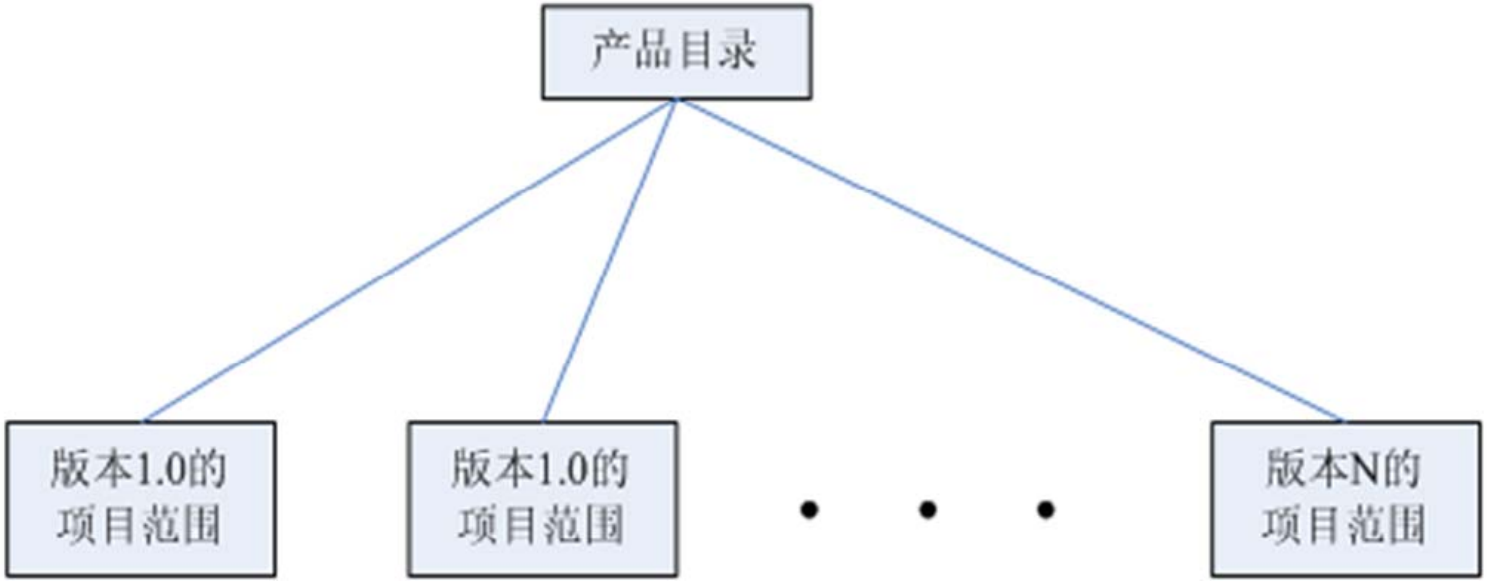
- 需求分析阶段的工作结果是获得高质量的具体软件需求。

软件需求分析的具体工作

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

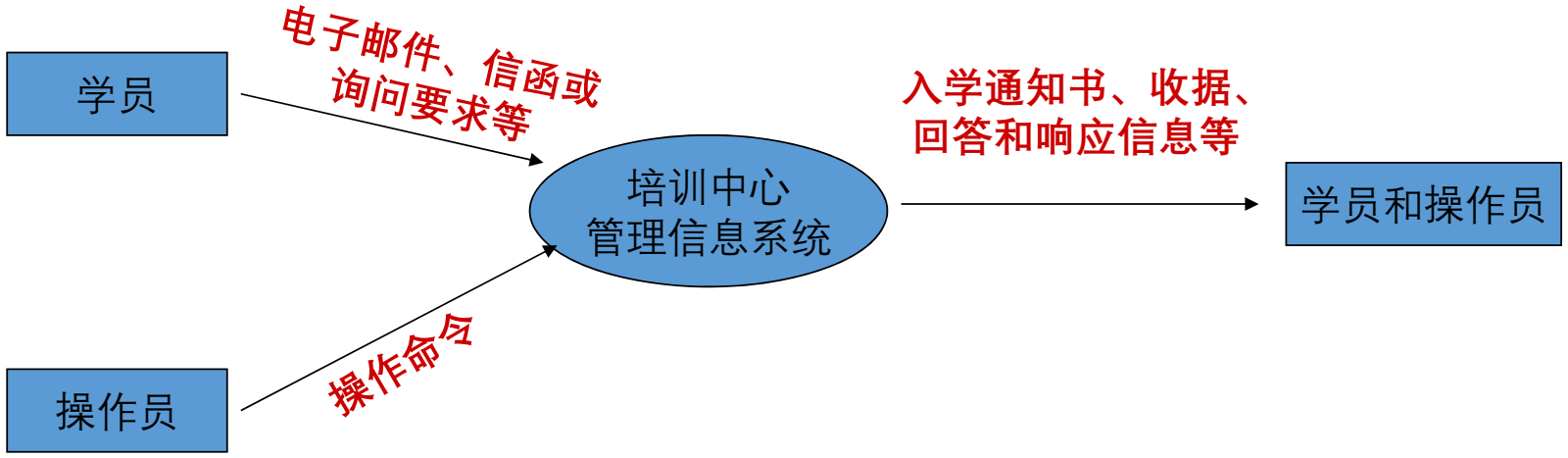
1. 建立系统关联图

- 建立系统关联图主要是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。
  - 项目目标：描述了产品用来干什么，它最终是什么样子（远期目标）
  - 项目范围：确定当前项目要解决项目长远规划中的哪一部分（近期目标）



1. 建立系统关联图

- 关联图是用于描述系统与外部实体间的界限和接口的模型，明确指出通过接口的信息流和物质流。
- 关联图不明确描述系统的内部过程和数据。
- 关联图以图形方式表示系统的范围使得项目相关人员更易于理解和审查。



2 分析需求的可行性

- 分析需求可行性的基本任务是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。
- 目的在于明确与每项需求相关联的风险，包括一些与其他方面的冲突、对外部环境的依赖和某些技术的障碍等。

在实际需求分析中应考虑的风险类型

- 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：实现这项需求可能导致无法满足整个系统的安全需求。
- 过程风险：实现这项需求可能导致需要对常规的开发过程做修改。
- 实现技术风险：实现这项需求可能需要使用不熟悉的实现技术。
- 数据库风险：实现这项需求可能导致系统不支持的非标准数据。
- 日程风险：实现这项需求可能遇到技术困难，并危及系统原定的开发日程。
- 外部接口风险：实现这项需求可能涉及外部接口。
- 稳定风险：这项需求可能是易变的，将导致开发过程的重大变动。

3 构建用户接口原型

- 创建用户接口原型的基本任务是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。
- 用户接口原型是指一个可能的局部实现，不是整个系统。
- 在需求分析中遇到具有不确定性、二义性、不完整或含糊特征的需求时，建议使用抛弃型原型。
- 在螺旋式开发模型中使用进化型模型。

构建用户接口原型的方法

- 纸上原型化方法
- 人工模拟原型化方法
- 自动原型化方法

4 确定需求的优先级别

- 划分优先级可以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。
- 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，帮助解决可能发生的设计冲突。
- 优先级的分配应当由软件开发人员和项目相关人员共同完成。

多种分配需求优先级的方法

命 名	含 义	方法来源
高	一个关键任务的需求；下一版本所需求的	Karl E. W.
中	支持必要的系统操作；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	一个功能类，实现或不实现均可	
3	必须完美地实现	（Kovitz 1999）
2	需要付出努力，但不必做得太完美	
1	可以包含缺陷 (但不是有意的☺)	

Bugzilla

- Blocker，阻碍开发和/或测试工作
- Critical，死机，丢失数据，内存溢出
- Major，较大的功能缺陷
- Normal，普通的功能缺陷
- Minor，较轻的功能缺陷
- Trivial，产品外观上的问题或一些不影响使用的小毛病，如菜单或对话框中的文字拼写或字体问题等等
- Enhancement，建议或意见

Bugzilla

- Highest:
  - Crash/Loss of data/Inability to install/Other broken core functionality. Affects all/nearly-all users and documents.
  - We reserve this level for our "Most Annoying Bugs" -- items that need attention super-promptly.
- High:
  - Serious problems/Inability to open certain documents/Tediously slow.
  - Affects many users.
- Medium:
  - Prevents users from making professional-quality work or breaks some features.
- Low:
  - Doesn't affect ability to make high-quality work. Suggestion for improvement.
- Lowest:
  - Wishlist item, Incredibly minor issue, or similar.

Bugzilla(Mozilla enhancement)

- P1 = We definitely want this. It's a major feature, and it's obvious that it would be useful to everybody.
- P2 = We want this, but it's not totally clear or extremely important.
- P3 = This isn't a bad idea, and maybe we'll want to implement it at some point in the future, but it's not near-term roadmap material. Some core Bugzilla developer may work on it.
- P4 = This isn't a terrible idea, but it's not important to our long-term plans for Bugzilla. We would review a patch if somebody posted it, but a core developer is unlikely to work on it. If the patch proves to be too complex, there's a chance that the feature will be marked WONTFIX or delayed until some unknown future release.
- P5 = We basically never want this. If somebody implements it and asks for review, we *might* look at it. If a posted patch involves any significant complexity, it will likely be rejected.

设定需求优先级

- 建议
  - 需求获取期间就关注需求优先级的问题
  - 基于用例标注优先级
  - 基于场景标注优先级，尤其对例外的处理。有些例外会对系统产生根本影响如system crash，有些只是轻微的。例外很难穷举，但配以优先级就容易裁决
  - 建立模型可以发现哪个功能更重要，排定优先级
  - 优先级并非在整个开发过程中一成不变的 – 需求管理方面的问题
  - 可以使用电子表单、表格或矩阵来估计和维护一个功能或用例的优先级

5 需求建模

- 需求建模的工作是导出系统的逻辑模型，以明确目标系统“做什么”的问题。
- 所谓模型，是为了理解事物而对事物作出的一种抽象，是对事物的一种无歧义的书面描述。
- 需求建模是指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。

多种需求分析（建模）方法

- 数据需求分析
- 行为需求分析
- 功能需求分析
- 非功能性需求分析
- 实时系统需求分析
- ……

需求分析方法

- 结构化分析方法
- 面向对象的需求分析方法

需求建模：结构化分析方法

- 结构化分析方法(Structured Analisys, SA)
  - 基于数据流技术的分析方法
  - 适用于数据处理类型软件的需求分析

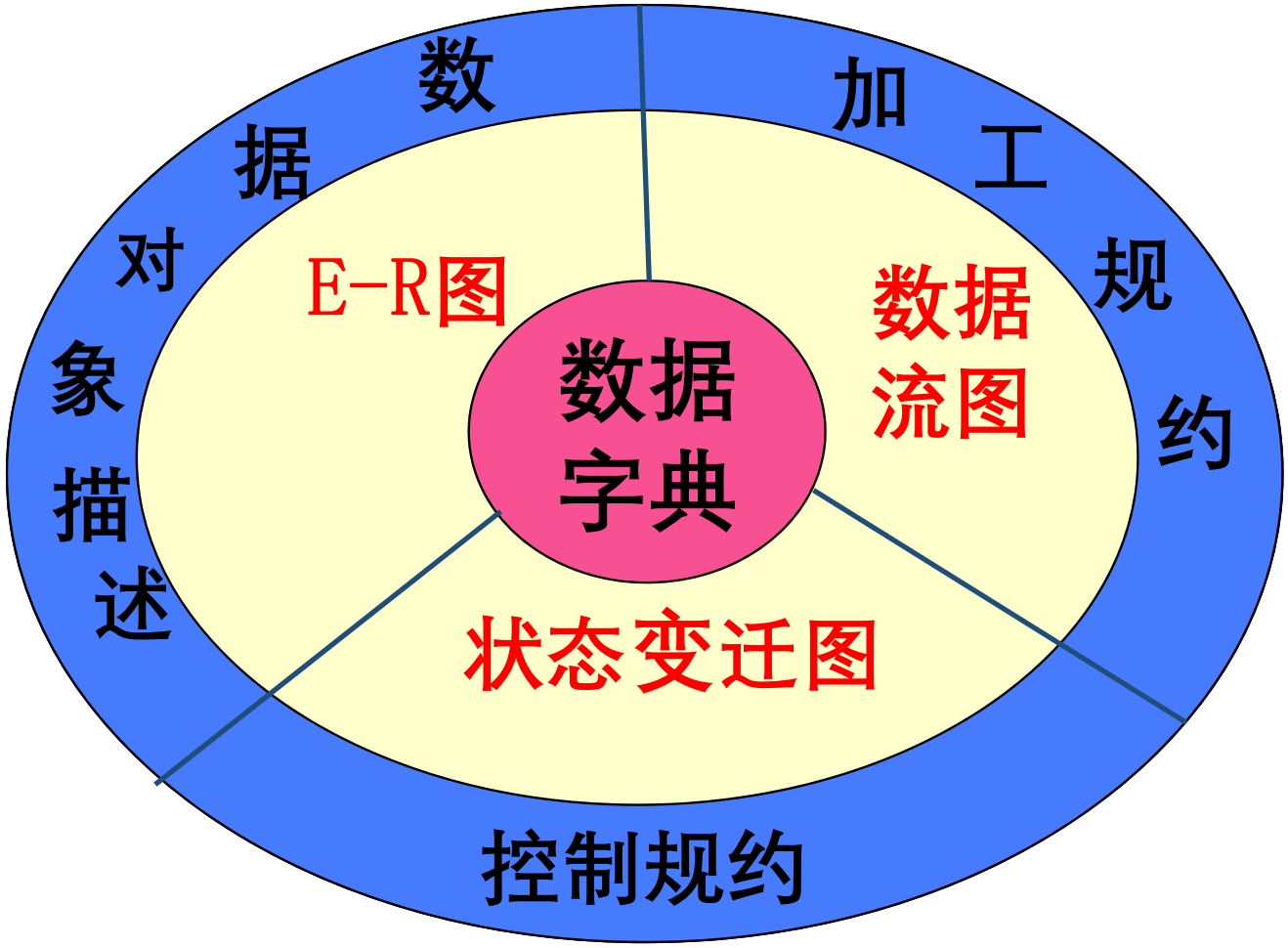
结构化方法分析基本步骤

1. 画出数据流图。设计数据流图必须逐步求精；
2. 决定哪些部分需要计算机化和怎样计算机化（取决于用户投资限制和自身技术限制）；
3. 描述数据流细节，大型软件可以使用数据字典描述所有数据元素；
4. 定义处理逻辑（加工逻辑：每个加工处理做什么）；
5. 定义数据存储，即定义每个存储的确切内容及其表示法（格式）；
6. 定义物理资源：如是文件需指定：文件名、组织结构（排序、索引等）、存储介质和记录；如是数据库需指定每个表的相关信息；
7. 确定输入输出规格说明，如输入内容、输入屏幕、打印输出格式、输出长度等等；
8. 确定硬件所需有关数值，如输入量、打印频率、CPU、记录大小、数据量大小、文件大小等等；
9. 确定软硬件接口和环境需求。

分析模型的主要目标

- 描述用户需要
- 建立创建软件设计的基础
- 定义软件完成后可被确认的一组需求

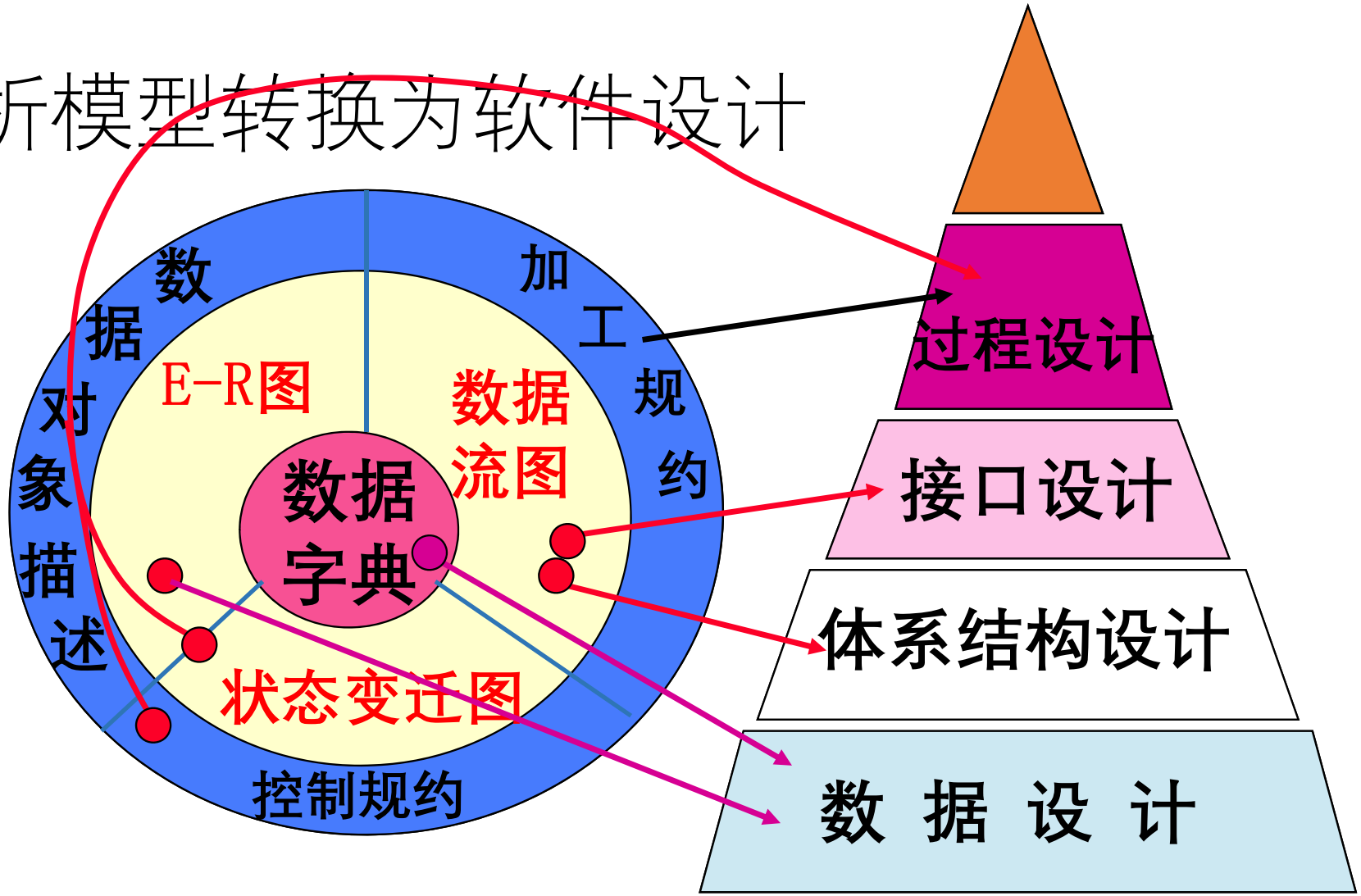
结构化分析模型的结构



分析模型的元素

- 数据字典(DD)：模型核心(中心库)
- E-R图(ERD)
- 数据流图(DFD)
  - 指明数据在系统中移动时如何被变换;
  - 描述对数据流进行变换的功能;
  - DFD中每个功能的描述包含在加工规约 (小说明)。
- 状态变迁图(STD)
  - 指明作为外部事件的结果,系统将如何动作。

将分析模型转换为软件设计



结构化分析方法的需求建模

- 数据建模
- 功能建模
- 行为建模

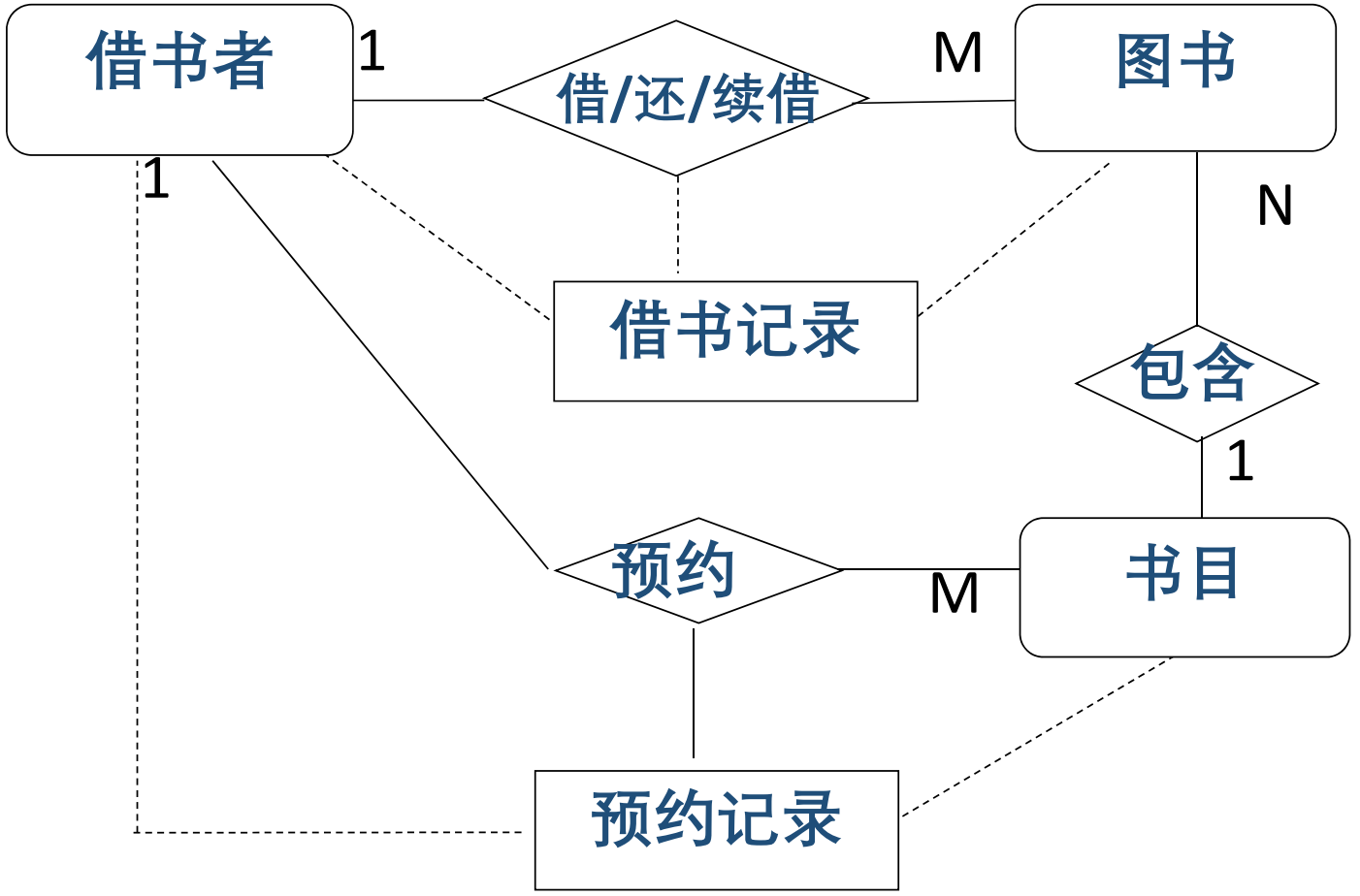
数据建模

E-R图是数据建模的基础

实例：图书馆系统的主要功能需求

- 图书借出：管理员完成一次借书过程。
- 图书归还：管理员完成一次还书过程。
- 图书预约：用户查询要借的图书，若不能借，可预约该图书。
- 图书续借：用户可以将图书的归还日期延长一段时间。
- 图书管理：添加新书。更新图书馆信息，销毁图书。
- 用户管理：注册新用户，更新用户信息，注销用户。
- 处理罚金：用户缴纳罚金吼，系统将罚金数额清零。

实例分析：图书馆系统



实例分析：图书馆系统

- 实体：图书、借书者、管理员、借书目录、预约记录、书目
- 属性给出如下：
  - 借书者：借书者编号、姓名、性别、借书数、最大借书数、罚金金额、有限期
  - 图书：图书号、书目号
  - 书目：书目号、书名、作者、出版社、丛书名、收藏数、在馆数、预约数
  - 借书记录：图书号、借书者编号、借出日期、应还日期、续借次数
  - 预约记录：书目号、借书者编号、预约日期

功能建模

- 使用抽象模型的概率，按照软件内部数据传递、变换的关系，自顶向下逐层分解、细化，直到找到满足功能需求的所有可实现的软件分量为止。
- 数据流图 DFD (Data Flow Diagram)
  - 描述逻辑模型的图形工具，表示数据在系统内的变化。
  - DFD可以用来表示一个系统或软件在任何层次上的抽象。
  - 较大型软件系统DFD分成多层(子图、父图概念)，可以表示数据流和功能的进一步的细节。

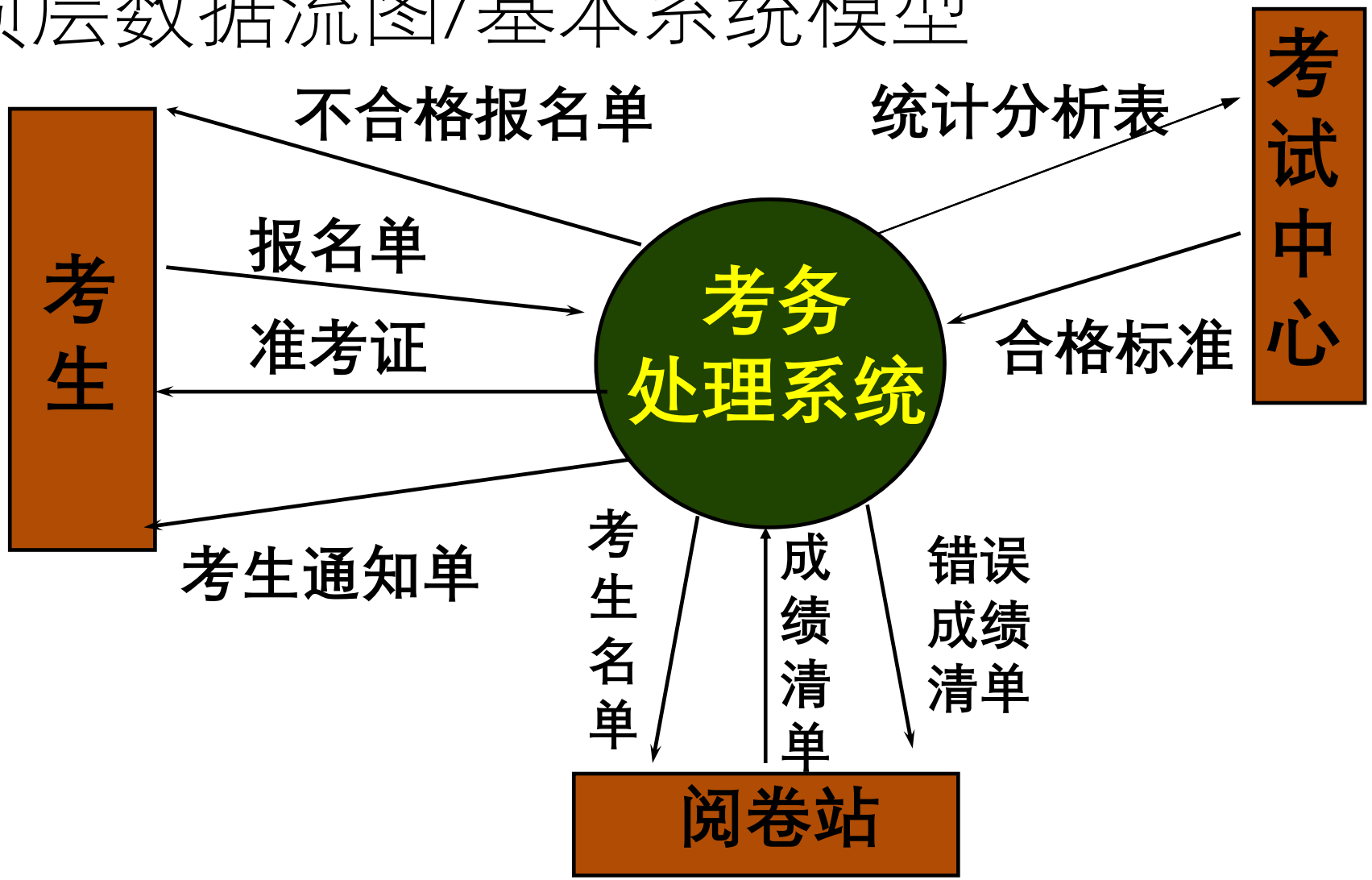
数据流图画法

1. 先找出系统的数据源点和汇点。
2. 找出与外部实体的输出数据流与输入数据流。
3. 在图的边上画出系统的外部实体。
4. 从外部实体的输出数据流出发，按照系统的逻辑需要，逐步画出一系列的逻辑加工，直到找到外部实体所需要的输入数据流，形成数据流的封闭。
5. 按照一定的原则进行检查和修改。
6. 画出所需要的子图。

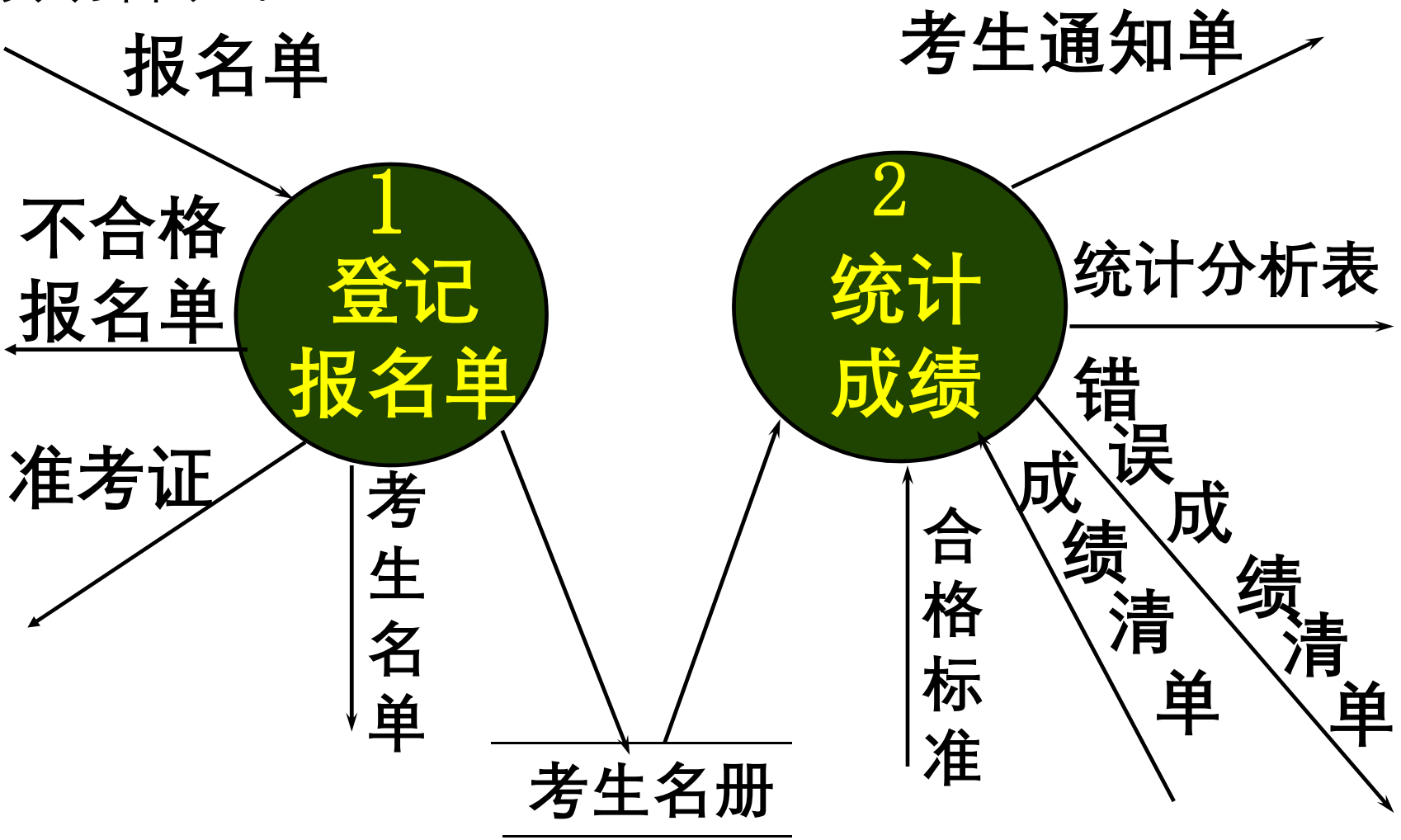
DFD实例：考务处理系统功能

1. 对考生送来的报名单进行检查;
2. 对合格的报名单编好准考证号后将准考证送给考生，并将汇总后的考生名单送给阅卷站;
3. 对阅卷站送来的成绩单进行检查，并根据考试中心制定的合格标准审定合格者;
4. 制作考生通知单(含成绩及合格/不合格标志)送给考生;
5. 按地区进行成绩分类统计和试题难度分析，产生统计分析表。

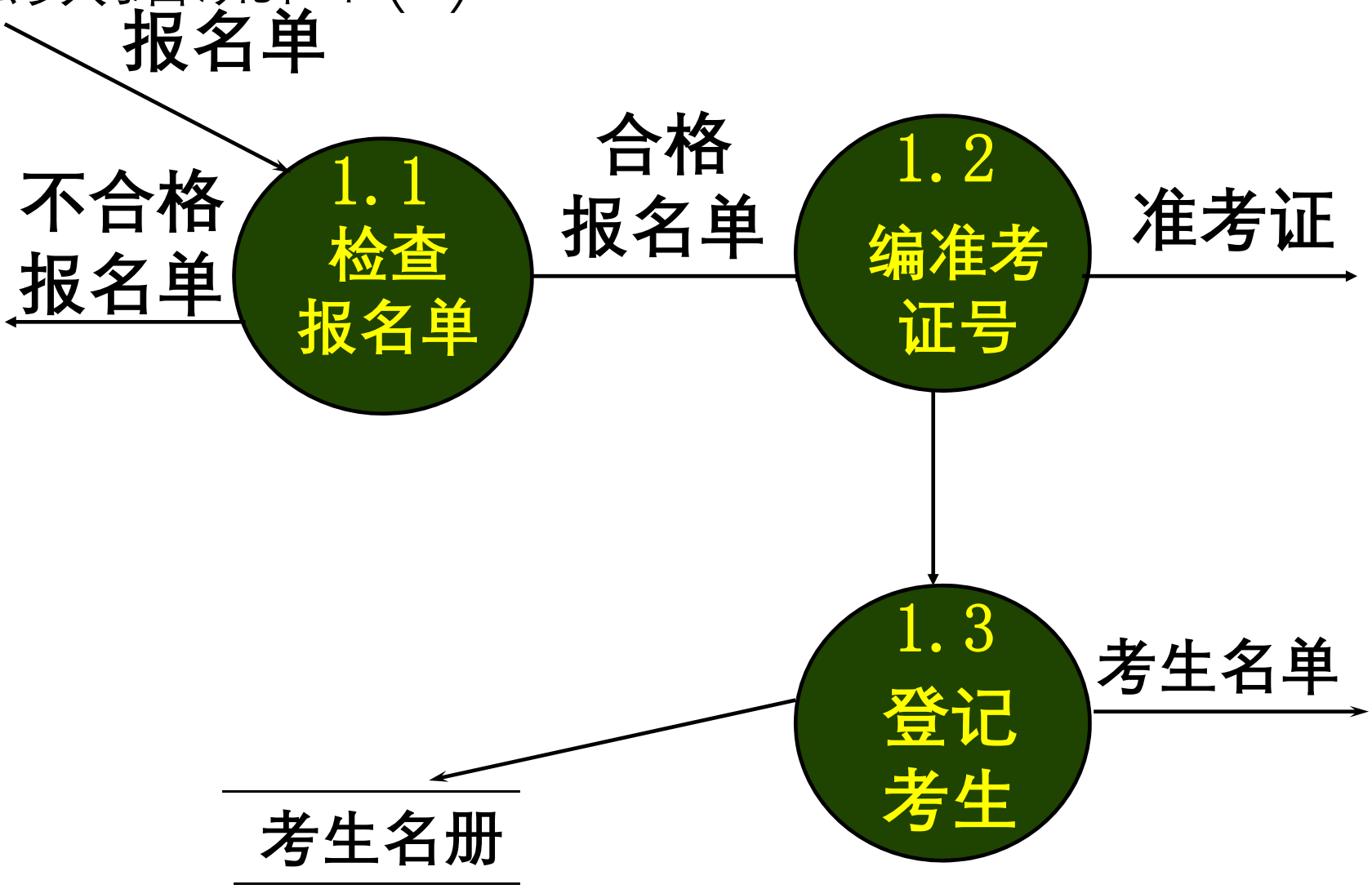
顶层数据流图/基本系统模型



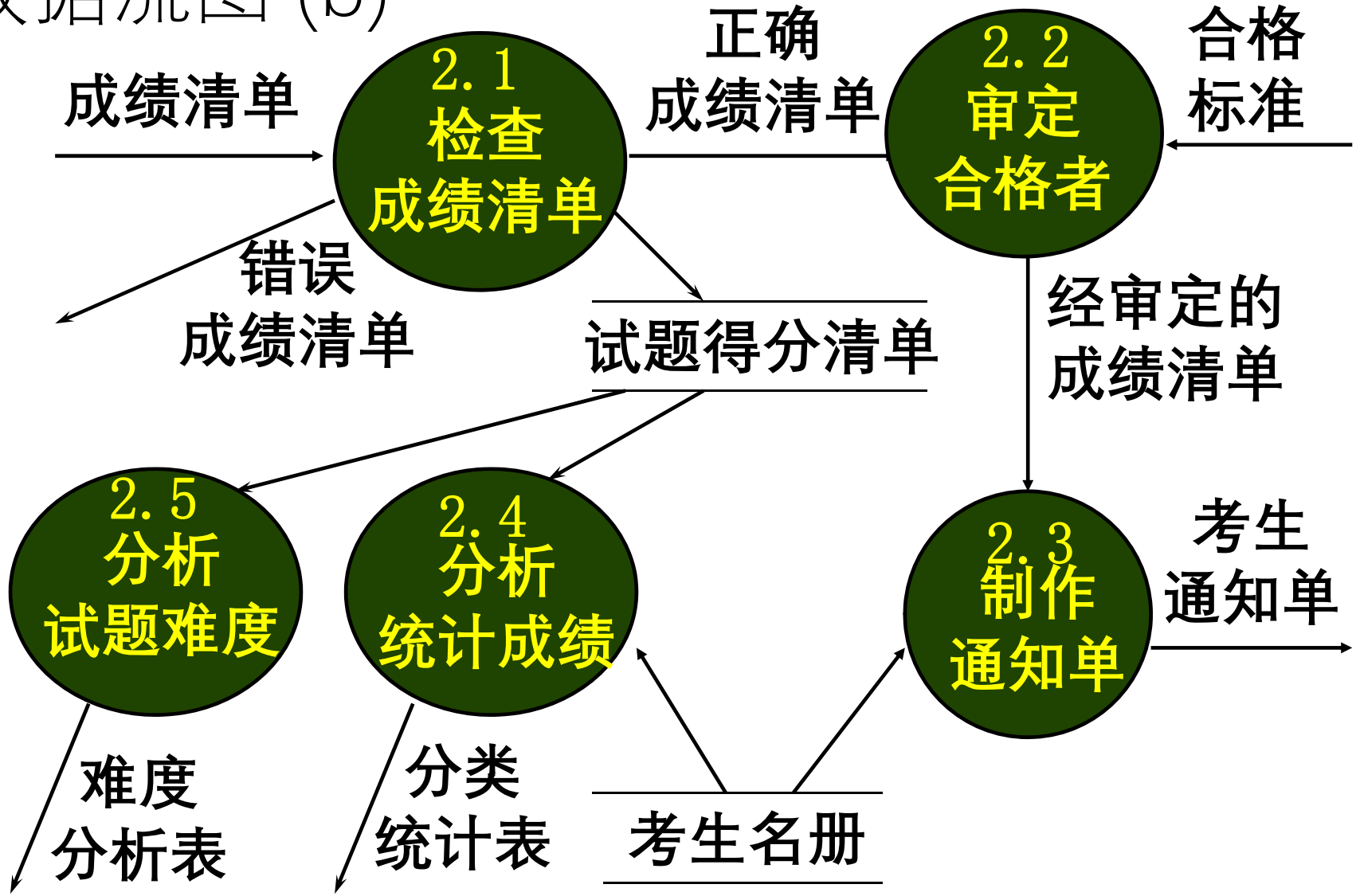
0层数据流图


报名单
考生通知单
1
登记
报名单
2
统计
成绩
不合格
报名单
统计分析表
准考证
考生
名单
考生名册
合格标准
错误成绩清单
成绩清单

一层数据流图 (a)



一层数据流图 (b)

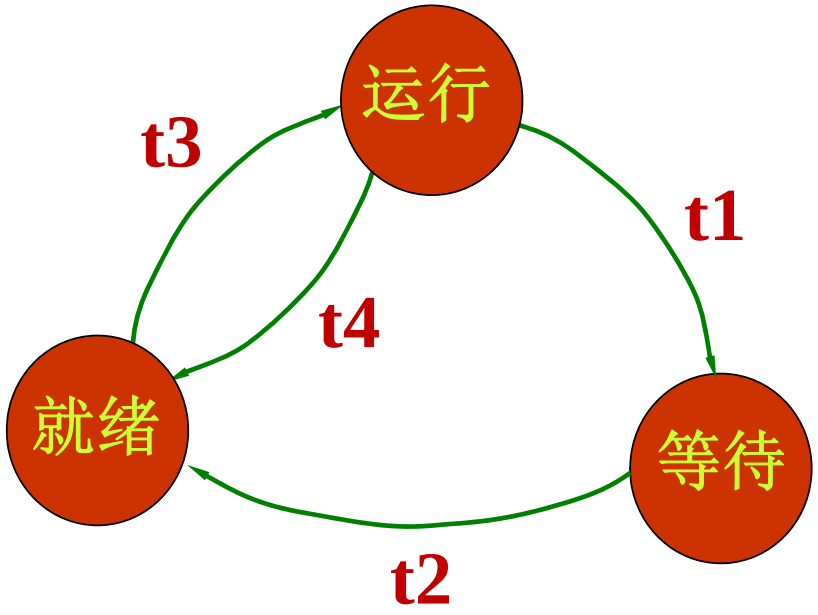

成绩清单
2.1
检查
成绩清单
正确
成绩清单
2.2
审定
合格者
合格
标准
错误
成绩清单
试题得分清单
经审定的
成绩清单
2.5
分析
试题难度
2.4
分析
统计成绩
2.3
制作
通知单
考生
通知单
难度
分析表
分类
统计表
考生名册

行为建模

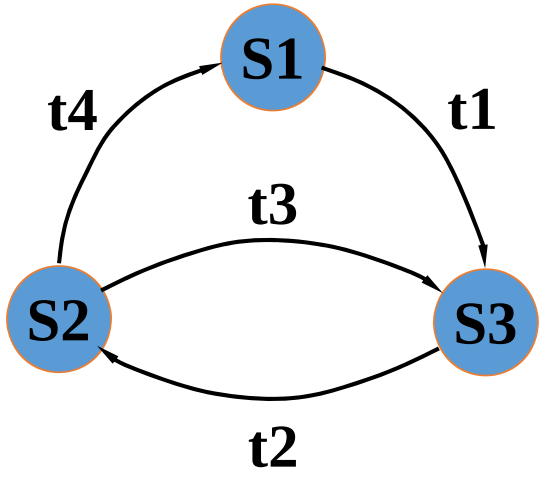
- 行为建模给出需求分析方法的所有操作原则，但只有结构化分析方法的扩充版本才提供这种建模的符号。
- 数据流图不描述时序关系，控制和事件流 通过行为模型描述。
- 在描述系统或各个数据对象的行为时，采用状态迁移图。通过描述系统或对象的状态，以及引起系统或对象状态转换的事件来表示系统或对象的行为。

状态迁移图

- 状态迁移图是描述系统的状态如何相应外部的信号进行推移的一种图形表示。
- 例如，有关处理器分配的进程状态迁移。



- 在状态迁移图中，
  - ✓ “○”表示可得到的系统状态
  - ✓ “→”表示从一种状态向另一种状态的迁移。
- 在箭头上要写上导致迁移的信号或事件的名字。



状态 / 事件	S1	S2	S3
t1	S3		
t2			S2
t3		S3	
t4		S1	

需求分析方法

- 结构化分析方法
- 面向对象的需求分析方法

面向对象的需求分析方法

- 面向对象方法都支持三种基本的活动：识别对象和类，描述对象和类之间的关系，以及通过描述每个类的功能定义对象的行为。
- 关注于定义类和以及类之间的协作方式
- 面向对象的需求建模方法的关键是从获取的信息中识别出问题域中的类和对象，并分析它们之间的关系，最终建立起简洁、精确和易理解的需求模型。

几种典型的面向对象方法简介

- Booch面向对象方法
- Jacoson的面向对象方法
- Coad-Yourdon的面向对象方法论
- OMT 方法(Object Modeling Technique)

1. Booch面向对象方法

- Booch面向对象方法
  - 一种迭代的、渐进的分析设计方法
- Booch 方法的基本开发模型
  - 开发模型包括逻辑模型、物理模型、静态模型和动态模型。所采用的对象模型要素是：封装、模块化、层次类型、并发。
- Booch方法可分为逻辑设计和物理设计
  - 逻辑设计包含类图文件和对象图文件
  - 物理设计包含模块图文件和进程图文件,用以描述软件系统结构。

- Booch方法的力量在于其丰富的符号体系，包括：
  . 类图（类结构－静态视图）
  . 对象图（对象结构－静态视图）
  . 状态转移图（类结构－动态视图）
  . 时态图（对象结构－动态视图）
  . 模块图（模块体系结构）
  . 进程图（进程体系结构）

2．Jacoson的面向对象方法

1. 建立面向对象分析模型的过程
    1. 建造用户需求模型。
    2. 建造系统分析模型。
2. 建立面向对象设计模型的过程
    1. 创建模块作为主要的设计对象。
    2. 创建一个显示激励如何在模块间传送的交互图。
    3. 把模块组织成子系统。
    4. 复审设计工作。

3．Coad-Yourdon的面向对象方法论

1. 面向对象分析（OOA）步骤
   - 类及对象层
   - 结构层
   - 主题层
   - 属性层
   - 服务层
2. 面向对象设计（OOD）步骤

4．OMT 方法(Object Modeling Technique)

- 由Loomis Shan和Rumbaugh在1987年提出的，曾应用于关系数据库的设计，1991年正式应用于面向对象分析和设计领域。

- 基本思想：围绕现实世界的客观实体（对象）与实体概念（问题域的问题）来构造系统模型，制定策略并逐步优化，直至趋于完善。
  - 对象模型
  - 动态模型
  - 功能模型

OMT方法的三种需求模型及其描述工具

- 对象模型
  - 定义了系统的静态结构。主要用来描述类或对象之间的静态关系。
  - 使用的描述工具：类图 对象图
- 动态模型
  - 定义了系统的动态结构。主要用来表达系统动态交互行为中对象的状态受消息影响而发生变化的时序过程。
  - 使用的描述工具 （主要对象的）状态图和序列图
- 功能模型
  - 定义系统应该做什么。主要用来描述系统从输入到输出的处理流程，描述与数据值的变化有关的系统属性。例如 功能 映射 约束等
  - 使用的描述工具：数据流图

课堂讨论：传统需求分析方法的应用

- 你能在AI-IDE项目中找到以上需求分析方法的痕迹吗？

- 面对AI-IDE项目的某个新需求，你能用以上需求分析方法对该需求进行建模吗？