CHAPTER 3

Software Requirement Analysis

3.3 需求获取的内容

(1) 功能
(2) 性能
(3) 环境
(4) 界面
(5) 接口
(6) 用户或人的因素
(7) 文档
(8) 数据
(9) 资源
(10) 安全保密
(11) 软件成本消耗与开发进度
(12) 质量保证

3.4 Requirement Elicitation （需求获取）

The process of discovering the client's requirement.

Called **requirement elicitation, or**

requirement discovering, or

requirement collection, or

requirement capture, or

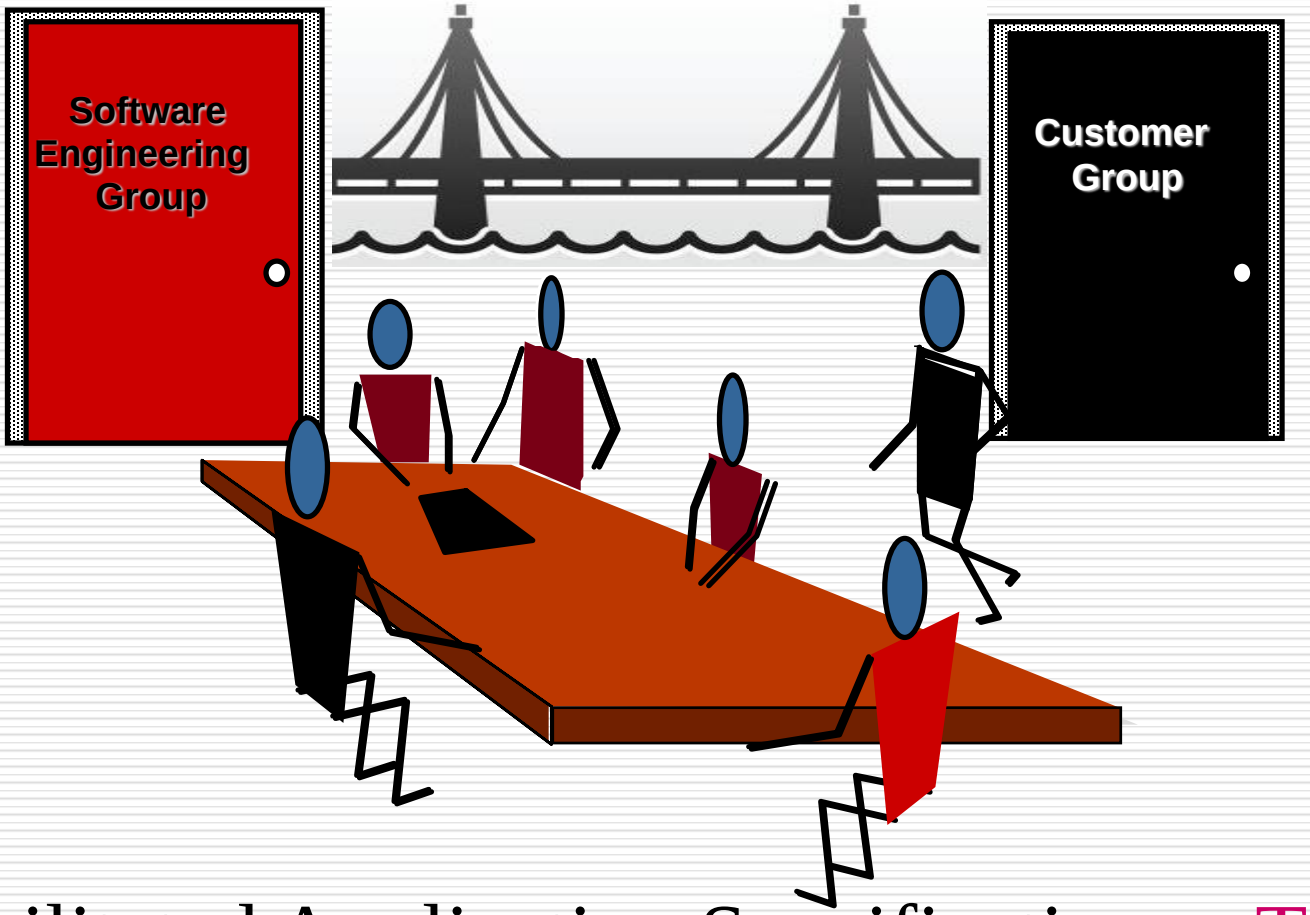
requirement gather

Requirements Elicitation (简介）

- 任务：主动与干系人协同工作，找出他们的需求，识别潜在的冲突，磋商解决矛盾，定义系统范围与边界。
- 关键：确保需求、该问题的解决是有商业价值的
- 注意以下问题：

✓ 产品设计目标不明确
✓ 干系人参与不足
✓ 干系人之间缺少共识
✓ 画蛇添足
✓ 需求快速变化
✓ 变更管理不足
✓ 需求分析不足

Requirements Gathering



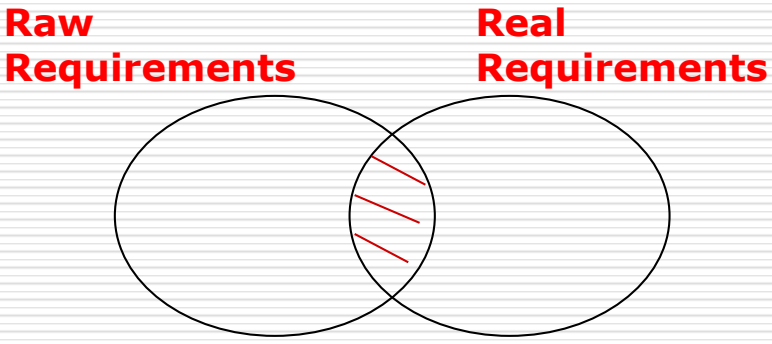
Facilitated Application Specification， Techniques

需求获取

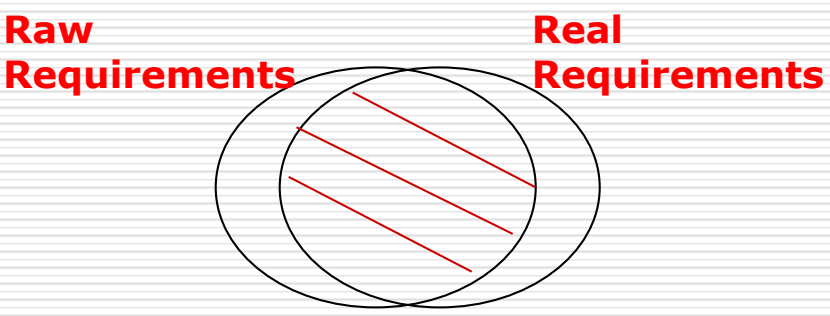
- **Stakeholders**
- **例子：HIS**

in individual or group requirements sessions to define system details

- ✓ **Results in a high quality system from the customer/user point of view**
- ✓ **Results in higher end-user satisfaction**
- ✓ **Helps to educate customer/user about system through their participation**



Without customer-oriented requirements-gathering practices



With customer-oriented requirements-gathering practices

需求获取

需求抽取过程中，最困难的不是记录用户需求，而是与用户探讨磋商，发现真正要解决的问题，确定适用的方案。

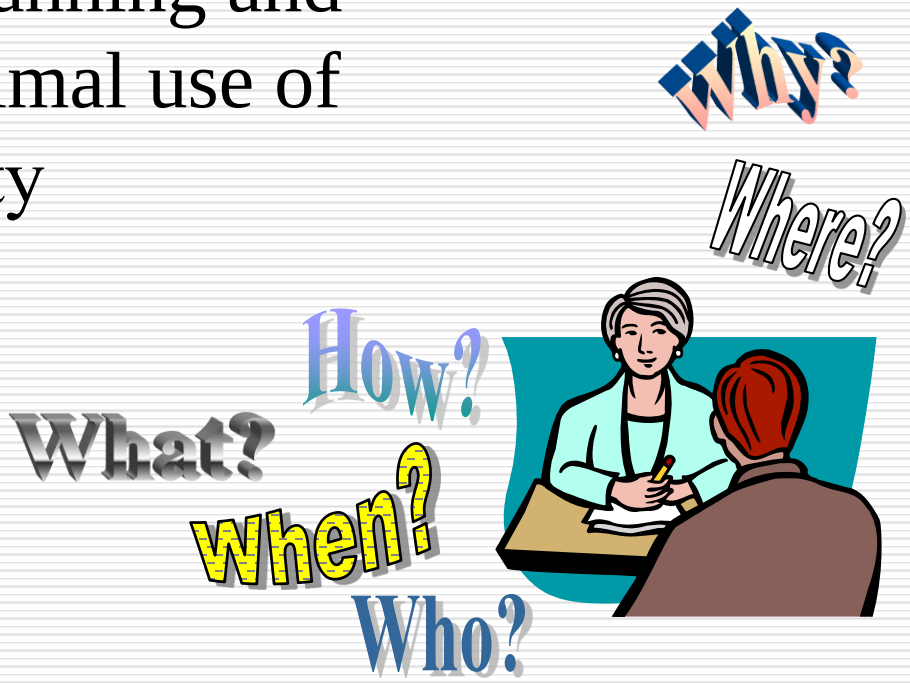
— Steve McConnell

Requirements Gathering(方法）

- Interview
- Scenarios
- Viewpoints
- Videotape camera
- Inquire table
- Web searching
- Requirements community in Internet like open source
- 闭门造车—格力电器—董明珠

(1) Interview 访谈，会谈

- Question and answer meetings in one-on-one or small group (focus group) setting
- May require detailed planning and facilitation to make optimal use of stakeholder's availability



Interview (最一般)

There are two types of interview

1. **Closed interviews where a pre-defined set of questions are answered.**
2. **Open interviews where there is no pre-defined agenda and a range of issues are explored with stakeholders.**

Interviews in practice

- Normally a mix of closed and open-ended interviewing.
- Interviews are good for getting an overall understanding of what stakeholders do and how they might interact with the system.
- Interviews are not good for understanding domain requirements
  - Requirements engineers cannot understand specific domain terminology;
  - Some domain knowledge is so familiar that people find it hard to articulate or think that it isn’t worth articulating.

Effective interviewers

- Interviewers should be open-minded, willing to listen to stakeholders and should not have pre-conceived ideas about the requirements.
- They should prompt the interviewee with a question or a proposal and should not simply expect them to respond to a question such as ‘what do you want’.

Before interview

- **Plan and schedule for interview**
  - ✓ **Define topic**
  - ✓ **Organize your questions.**
  - ✓ **Designate roles to team members.**
  - ✓ **Inform the client.**
  - ✓ **Indicate the topic.**

Before interview

- **Prepare for interview**
  - ✓ **learn about the client's business area.**
  - ✓ **learn about the client's organization.**
  - ✓ **memorize client's names.**
  - ✓ **be familiar with application domain**
  - ✓ **be familiar with right terminology**
  - ✓ **build a glossary**
  - ✓ **open questions in freedom, why? when? where?**
  - ✓ **preplanned question----form, check list**

During interview

- **Open interview**
  - ✓ **Introduce team.**
  - ✓ **Summarize previous meetings findings (if applicable).**
  - ✓ **Introduce topics.**
- **Body of interview**
  - ✓ **Make clients feel they are participants in the solution.**
  - ✓ **Take notes, but listen.**
  - ✓ **Keep it short.**
  - ✓ **Keep it focused.**
- **Close interview**
  - ✓ **Summarize.**
  - ✓ **Thanks clients for their time.**

After interview

- ✓ Immediately organize your notes
- ✓ Summarize findings
- ✓ Identify points still unclear

  (starting point for next)
- ✓ Evaluate your performance
- ✓ ……

(2) Scenarios (讲故事，形象，生动)

- ✓ **story, film， real-life rather than abstraction**
- ✓ **a storyboard, a serious of diagrams depicting the sequence of actions or events, a paper prototype**
- ✓ **that is, a serious of sheet of paper depict screen and user's response**
- ✓ **Eg. ATM card，支付宝**
- ✓ **use case in object oriented analysis**

Scenarios

- Scenarios are descriptions of how a system is used in practice
- They are helpful in requirements elicitation as people can relate to these more readily than abstract statement of what they require from a system
- Scenarios are particularly useful for adding detail to an outline requirements description

(3) Viewpoints （观点，视角，干系人）

- **Viewpoints are a way of structuring the requirements to represent the perspectives of different stakeholders. Stakeholders may be classified under different viewpoints.**
- **This multi-perspective analysis is important as there is no single correct way to analyse system requirements.**

Types of viewpoint

- **Interactor viewpoints**
  - ✓ **People or other systems that interact directly with the system. In an ATM, the customer's and the account database.**
- **Indirect viewpoints**
  - ✓ **Stakeholders who do not use the system themselves but who influence the requirements. In an ATM, management and security staff are indirect viewpoints.**
- **Domain viewpoints**
  - ✓ **Domain characteristics and constraints that influence the requirements. In an ATM, an example would be standards for inter-bank communications.**

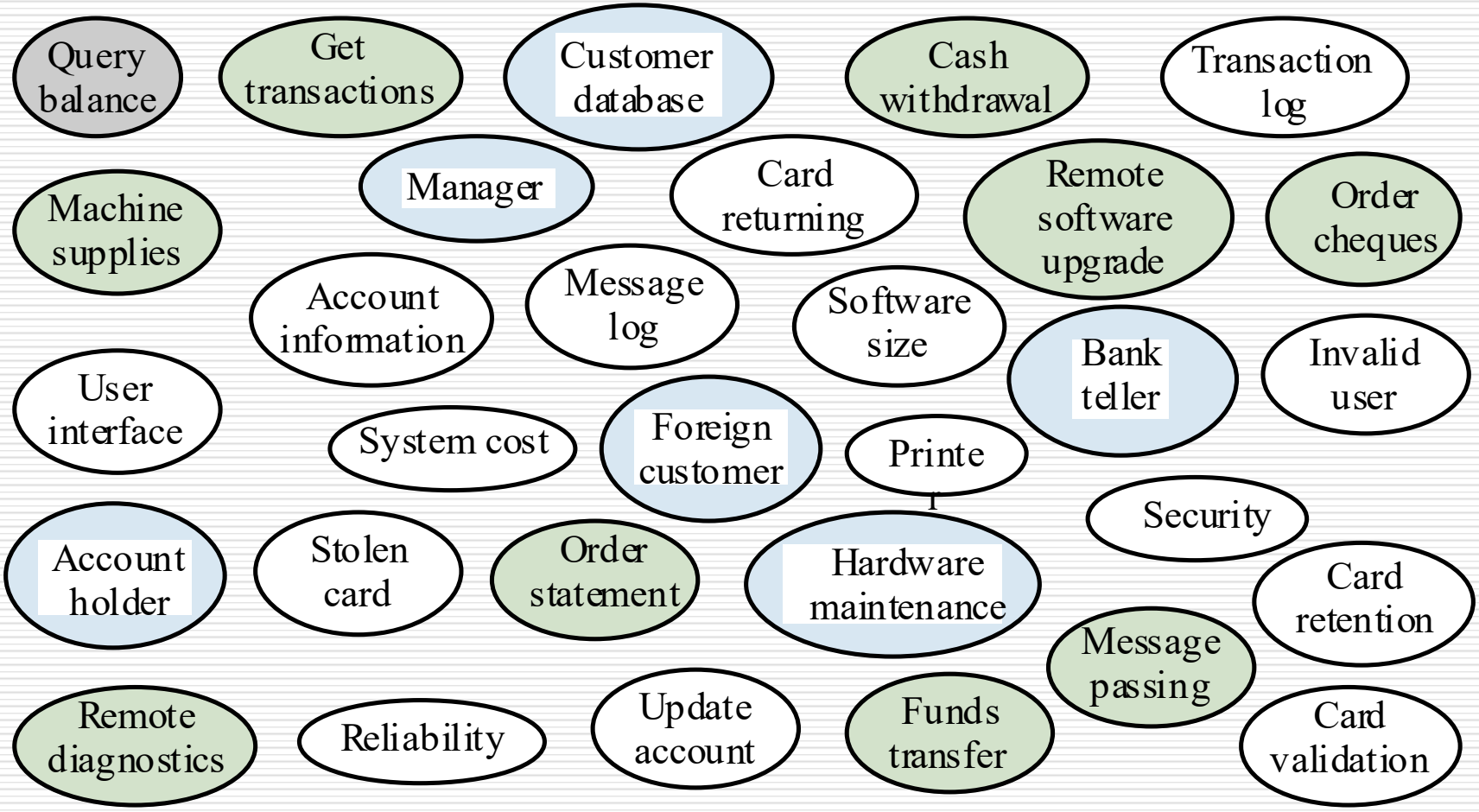
Viewpoint identification

- **Identify viewpoints using**
  - ✓ **Providers and receivers of system services;**
  - ✓ **Systems that interact directly with the system being specified;**
  - ✓ **Regulations and standards;**
  - ✓ **Sources of business and non-functional requirements.**
  - ✓ **Engineers who have to develop and maintain the system;**
  - ✓ **Marketing and other business viewpoints.**

An example: ATM viewpoints

- ✓ **Bank customers**
- ✓ **Representatives of other banks**
- ✓ **Hardware and software maintenance engineers**
- ✓ **Marketing department**
- ✓ **Bank managers and counter staff**
- ✓ **Database administrators and security staff**
- ✓ **Communications engineers**
- ✓ **Personnel department**
- ✓ **……**

Viewpoint identification



(4) Videotape camera

- ✓ **set up videotape cameras within workplace to record exactly what is being do.**
- ✓ **play back the tape to obtain requirement information.**
- ✓ 仿真，仿制，模仿，学习引进先进系统

xxxx年秋季入学新生报到、注册现场部分图片

学生先领取录取通知书、学籍卡片、学生证

学生在老师指导下填写学生证、卡片

学生交费后统一换取正式发票

学生办完手续后，买取教材

学生凭发票交回卡片，领取光盘、学生手册、盖章

(5) questionary，inquire table

（调查表，问讯表）

某出版社系统调查表

编号	提出问题
1	您在哪个部门工作？
2	出版业务流程是什么？
3	您每日都处理那些文件、数据、报表？
4	工作中手工处理特别麻烦的事情是什么？
5	工作中手工处理什么问题解决不了？影响效率的问题有哪些？
6	您认为提高工作效率，节省工作时间，减轻工作强度可采取哪些办法？

某出版社系统调查表

编号	提出问题
7	您的部门需要成本核算和统计的内容有哪些？
8	您的部门采用计算机管理工作情况如何？
9	如何改进业务流程使之更合理？
10	哪些问题是目前传统手工方法根本无法解决的？
11	出版社计算机管理信息系统需要解决什么问题？

(6) Web searching

- 互联网是最聪明的老师
  - ✓ 海量的信息库
  - ✓ 人类的知识库
  - ✓ 万能的Web
  - ✓ 向互联网学习，找答案
  - ✓ ……

(7) 网上需求社区

- 正如开源社区一样
- 通过开放讨论，共同构思需求，共享提出的需求
- 例如：
  - ✓ 未来手机的需求
  - ✓ 智能衣服的需求
  - ✓ 智能家居的需求

Requirements Gathering(技术比较）

- **Interviews**
  - Same Place, Same Time
  - Few People, Analyst-Driven
- **Questionnaires**
  - Different Time, Different Place
  - Many People, Analyst-Observer
- **Group Sessions**
  - Same or Different Place, Same Time
  - <20 People, Analyst-Facilitated
- **Observation**
  - Same Time, Same Place
  - Analyst-Observer

- Documentation Review
- Brainstorming
- Apprenticing
- Scenario Analysis
- Prototyping/Mock-up
- Modeling
- Workshops

A user requirement must be …

- ✓ documented
- ✓ expressed precisely
- ✓ expressed as what, not how
- ✓ prioritized
  - essential, desirable, optional
  - primary, secondary, tertiary
- ✓ testable
- ✓ covered in the design
- ✓ implemented
- ✓ traceable (in order to assure the last 3)

什么是高质量的用户需求？

- 真实性：真实地反映利益相关方的需求
- 一致性：需求项内部、需求项之间没有逻辑冲突
- 精确性：需求的表述不会引起二义或者多义理解
- 无冗余：每项需求在软件需求模型中仅出现一次，多项需求之间不存在语义重叠
- 完全性：所有需求项构成的全集完整地覆盖所有利益相关方的需求，尤其不能遗漏重要或紧迫的需求
- 可行性：在实际资源约束条件下，需求能够被完整实现的可能性
- 可验证性：在验收测试阶段，通过呈现测试结果，能够客观地、无争议地向利益相关方表明需求已完整实现

发现软件需求描述的缺陷

火车自动售票机控制软件的需求描述：

- 当用户点击“开始”按钮，包含所有潜在目的地的菜单被激活。
- 在用户选择目的地和乘车日期之后，系统要求用户插入银行卡。系统检查银行卡的有效性，并要求用户插入身份证。
- 在验证身份证的有效性之后，系统从信用卡扣款并快速弹出车票。

在真实性方面是否存在缺陷？

- 当用户点击“开始”按钮，包含所有潜在目的地的菜单被激活
  - 用户并不关心是否必须通过一个具体的“开始”按钮来启动购票流程
- 在用户选择目的地和日期后，系统要求用户插入银行卡。系统检查银行卡的有效性，并要求用户插入身份证
  - 购票时只需提供身份证号即可

一致性？

- 在用户选择目的地和日期后，系统要求用户插入银行卡。系统检查银行卡的有效性，并要求用户插入身份证。
- 在验证身份证的有效性之后，系统从信用卡扣款并快速弹出车票。

  - 除信用卡之外，是否需要支持具有支付功能的其他银行卡？

精确性？

- 当用户点击“开始”按钮，包含所有潜在目的地的菜单被激活。
  - 同城多站
- 在用户选择目的地和日期后，系统要求用户插入银行卡。系统检查银行卡的有效性，并要求用户插入身份证。
  - 银行卡的有效性有两种含义：(1)未被冻结，可正常使用；(2)除前一含义外，还要求余额足够支付票款
- 在验证身份证的有效性之后，系统从信用卡扣款并快速弹出车票。

完全性？

- 当用户点击“开始”按钮，包含所有潜在目的地的菜单被激活。
- 在用户选择目的地和日期之后，系统要求用户插入银行卡……
  - 必须支持用户手工输入目的地，并且系统动态地根据用户输入筛选潜在的目的地
  - 如果存在多条乘车路线，怎么办？

可行性？

- 当用户点击“开始”按钮，包含所有潜在目的地的菜单被激活。
  - 全国所有潜在的目的地的数量过于庞大！

可验证性？

- 在验证身份证的有效性之后，系统从信用卡扣款并快速弹出车票。
  - 不能客观地验证是否“快速”

改进的需求描述

- 用户输入目的车站的名字，系统动态地显示匹配的目的车站的清单。
- 在用户从清单中选定目的车站和乘车日期之后，系统给出所有可能的乘车路线及每条路线的车票价格。
- 用户从中选定一条路线之后，系统显示车票价格并提示用户插入银行卡、输入身份证号。系统也可以接收来自身份证识别装置传入的身份证号。
- 系统检查银行卡及身份证号的有效性，检查银行卡余额是否足够支付票款。检查通过后，系统从银行卡扣款并在用户输入齐备后2秒之内弹出车票。

Initial requirements

An example

Case Study: SafeHome

Product description:

Our research indicates that the market for home security systems is growing at a rate of 40% per year. We would like to enter this market by building a microprocessor-based home security system that would protect against and/or recognize a variety of undesirable "situations" such as illegal entry, fire, flooding, and others. The product, tentatively called *SafeHome*, will use appropriate sensors to detect each situation, can be programmed by the homeowner, and will automatically telephone a monitoring agency when a situation is detected. …

Case Study: SafeHome (Cont.)

Objects

-- smoke detectors, window and door detectors, motion detectors, an alarm, an event, a control panel a display, telephone numbers, a telephone call, …

Services

-- setting the alarm, monitoring the sensors, dialing the phone, programming the control panel, reading the display

Constraints

-- manufactured cost of less than $80, user-friendly, interfacing with a standard phone line

Performance Criteria

-- a sensor event recognized within one second

Case Study: SafeHome (Cont.)

Example of mini-specifications (object: control panel)

-- mounted on wall

-- size approximately 9 X 5 inches

-- contains standard 12-key pad and special keys

-- contains LCD display of the form shown in sketch

-- all customer interaction occurs through keys

-- used to enable and disable the system

-- software provides interaction guidance, echoes, and the like

-- connected to all sensors

原始需求难理解，正确性不能保证

原始需求与软件设计有什么关系？

Requirements validation techniques

- **Requirements auditing**
  - ✓ **Systematic manual analysis of the requirements.**
- **Prototyping**
  - ✓ **Using an executable model of the system to check requirements.**
- **Test-case generation**
  - ✓ **Developing tests for requirements to check testability.**
- **Modelling……**

3.5 Model and Modeling

- ✓ **Once an initial requirements has been determined, next step is to analysis and refine them (时机)**
- ✓ **Modeling help to analysis user requirements and change user requirement into software requirement.**
- ✓ **Model is also a part of specification from different perspectives and is a bridge for design.**

模型和建模

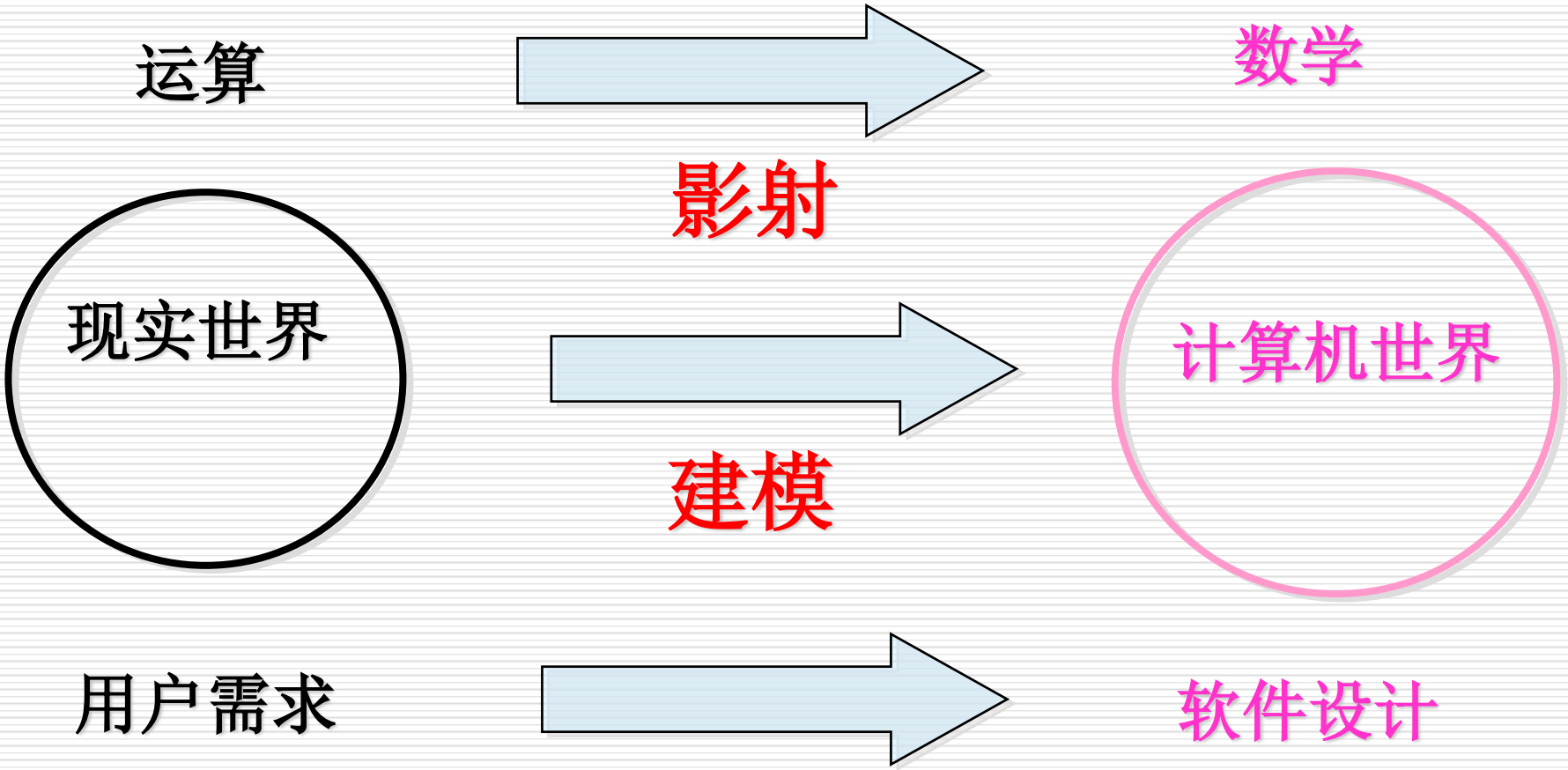
模型是现实世界某些重要方面的表示。

模型一种抽象，从某个视点、在某种抽象层次上详细说明被建模的系统。

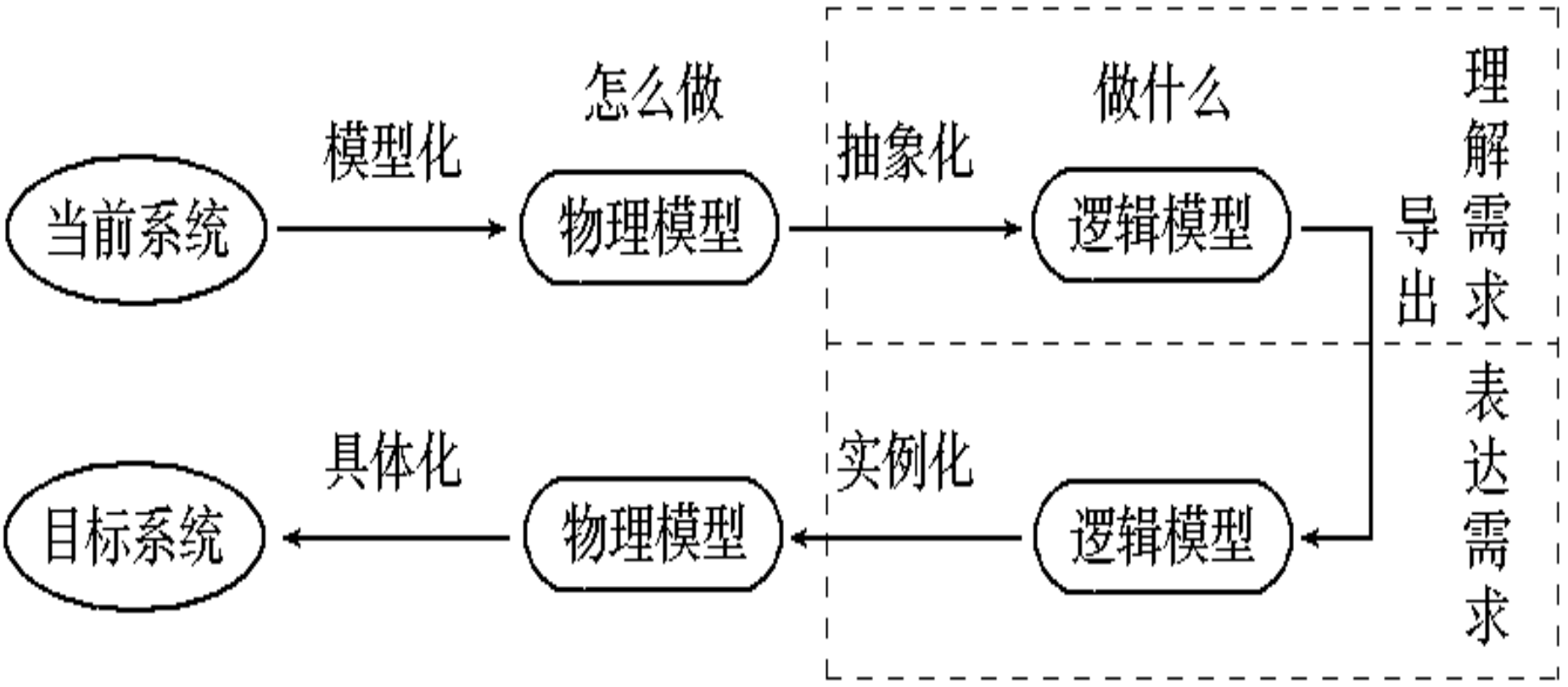
有时我们使用术语“**抽象**”来表示模型，因为我们从现实世界中抽象出对我们特别有用的东西。

模型一般分为具体模型和抽象模型两大类。具体模型有直观模型、物理模型等；抽象模型有思维模型、符号模型、图形模型、描述模型、数学模型等。

How to get model? 建模，抽象化……



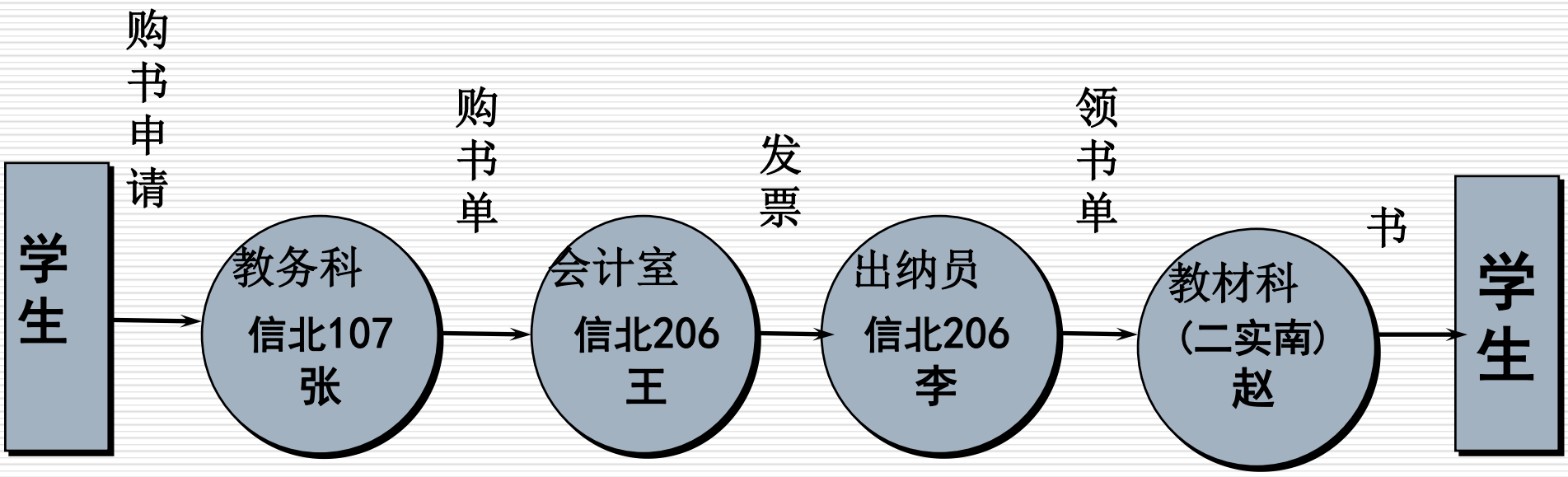
How to get model? 建模，抽象化……



建模分析过程 — 举例

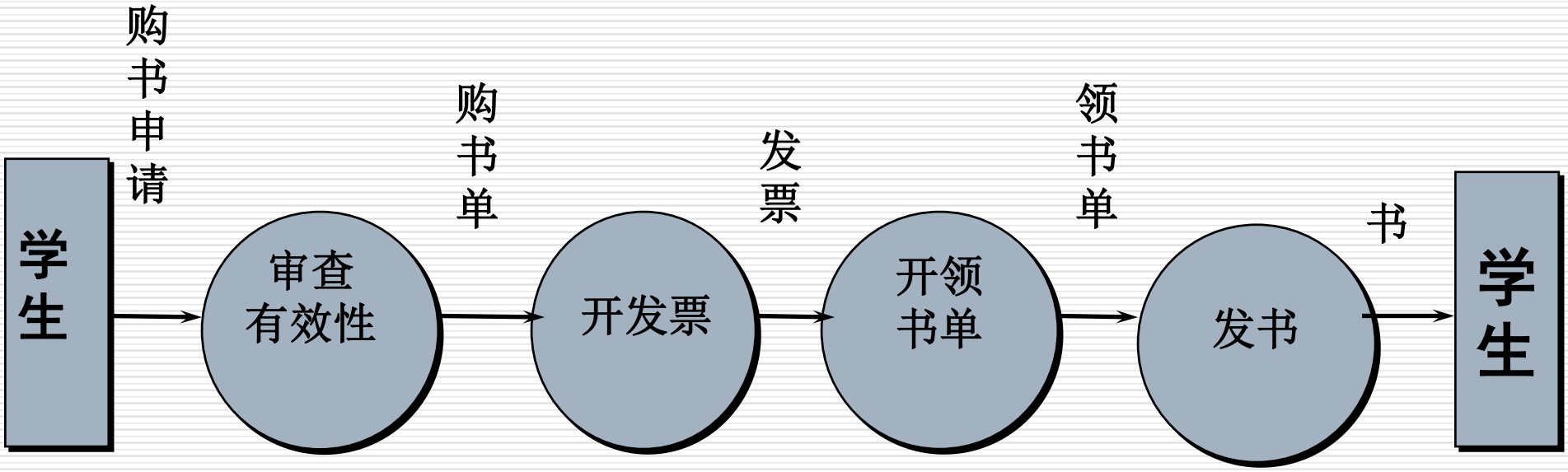
学生购买教材的实际处理流程 — 当前系统物理模型

（1）通过对现实环境的调查，获得当前系统的物理模型



建模分析的过程

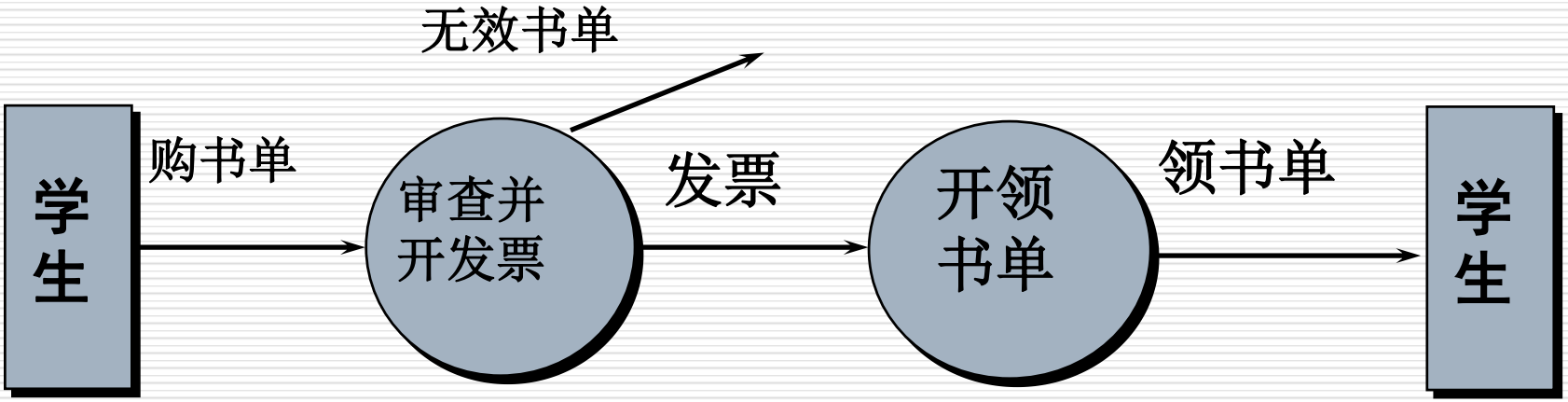
(2) 去掉具体模型中的非本质因素，抽取现实系统的实质，抽象出当前系统的逻辑模型。



学生购买教材的逻辑模型

建模分析的过程

(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



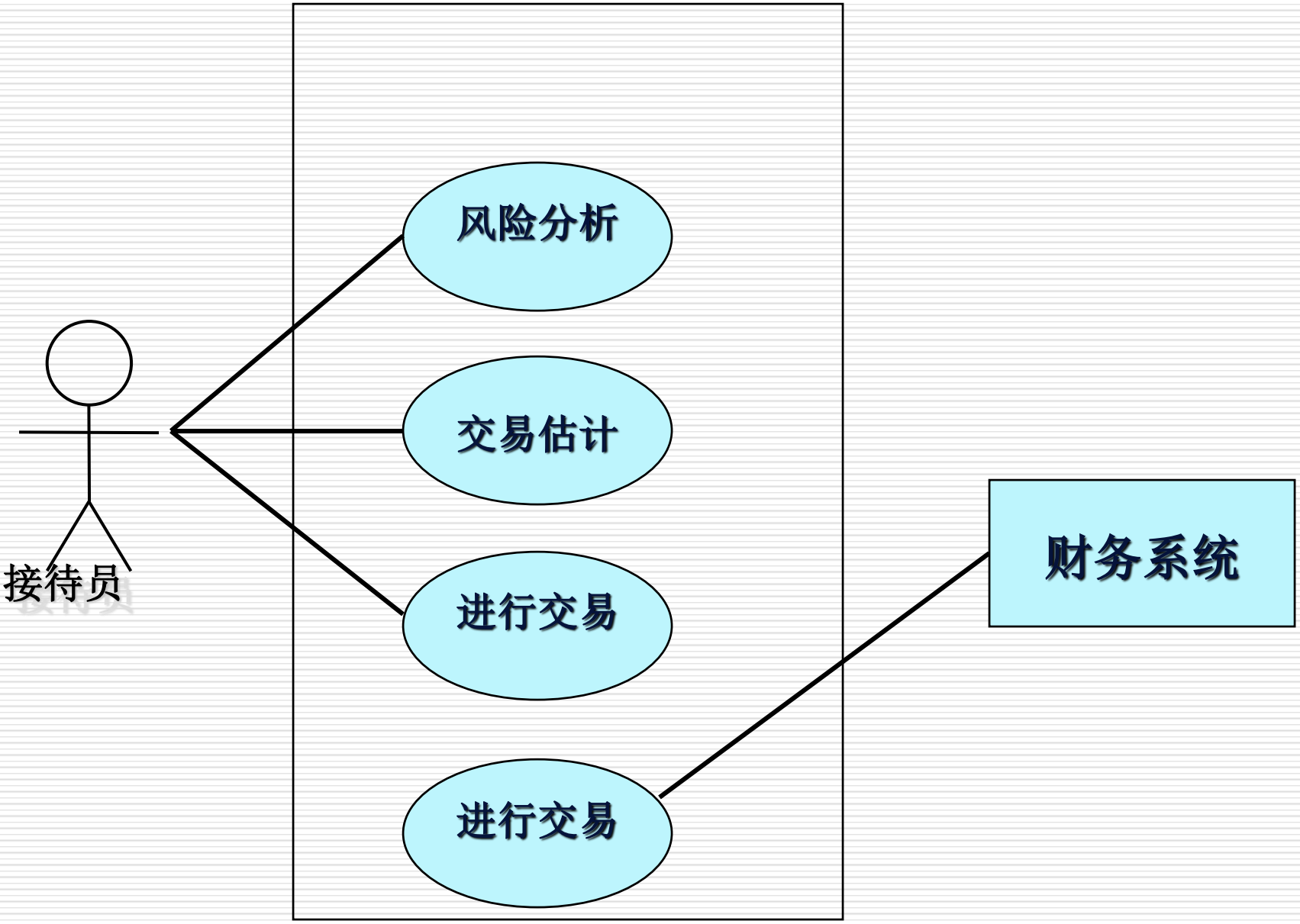
计算机教材管理系统的逻辑模型

建模分析过程示意

(4) 对目标系统的逻辑模型进行改进与优化

(5) 需求分析的验证

需求建模实例：某金融贸易系统用例图



总结：模型的作用

- 在建模过程中了解系统
- 通过抽象降低复杂性
- 有助于回忆所有的细节
- 有助于开发小组间的交流
- 有助于与用户的交流
- 为系统的维护提供文档

模型的作用

- ✓ 模型化或模型方法是通过抽象、概括和一般化，把研究的对象或问题转化为本质（关系或结构）相同的另一对象或问题，从而加以解决的方法。
- ✓ 模型化方法要求所建立的模型能真实反映所研究对象的整体结构、关系或某一过程、某一局部、某一侧面的本质特征和变化规律。

软件需求分析模型

- 数据流图（DFD）
- 实体—联系图（ **ERD** ）
- 类图
- 实例图
- 时序图
- 状态图
- 协作图
- 事件列表
- 数据字典模型
- 数据元素定义
- ……

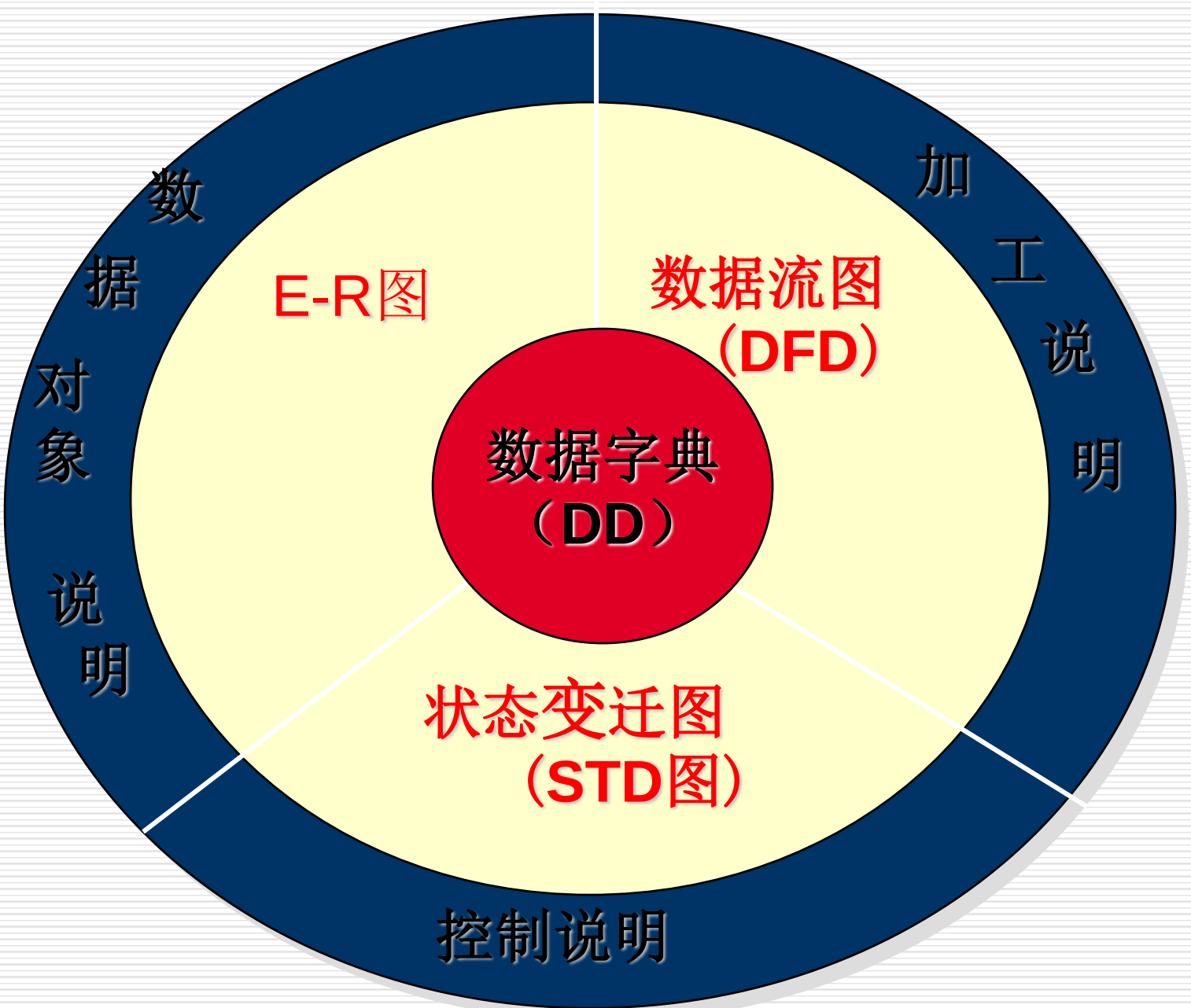
§3.6 软件需求分析建模方法

- 结构化分析(传统建模方法)
- 面向对象分析

§3.6.1 结构化分析方法

(Structured Analisys, SA)

结构化分析模型的组成结构



将分析模型转换为软件设计