需求工程过程

- 发现、分析和验证系统需求的过程

目标

- 描述主要的需求工程活动
- 介绍需求导出和分析的技术
- 描述需求有效性验证
- 讨论需求管理的地位，以及对其它需求工程过程的支持

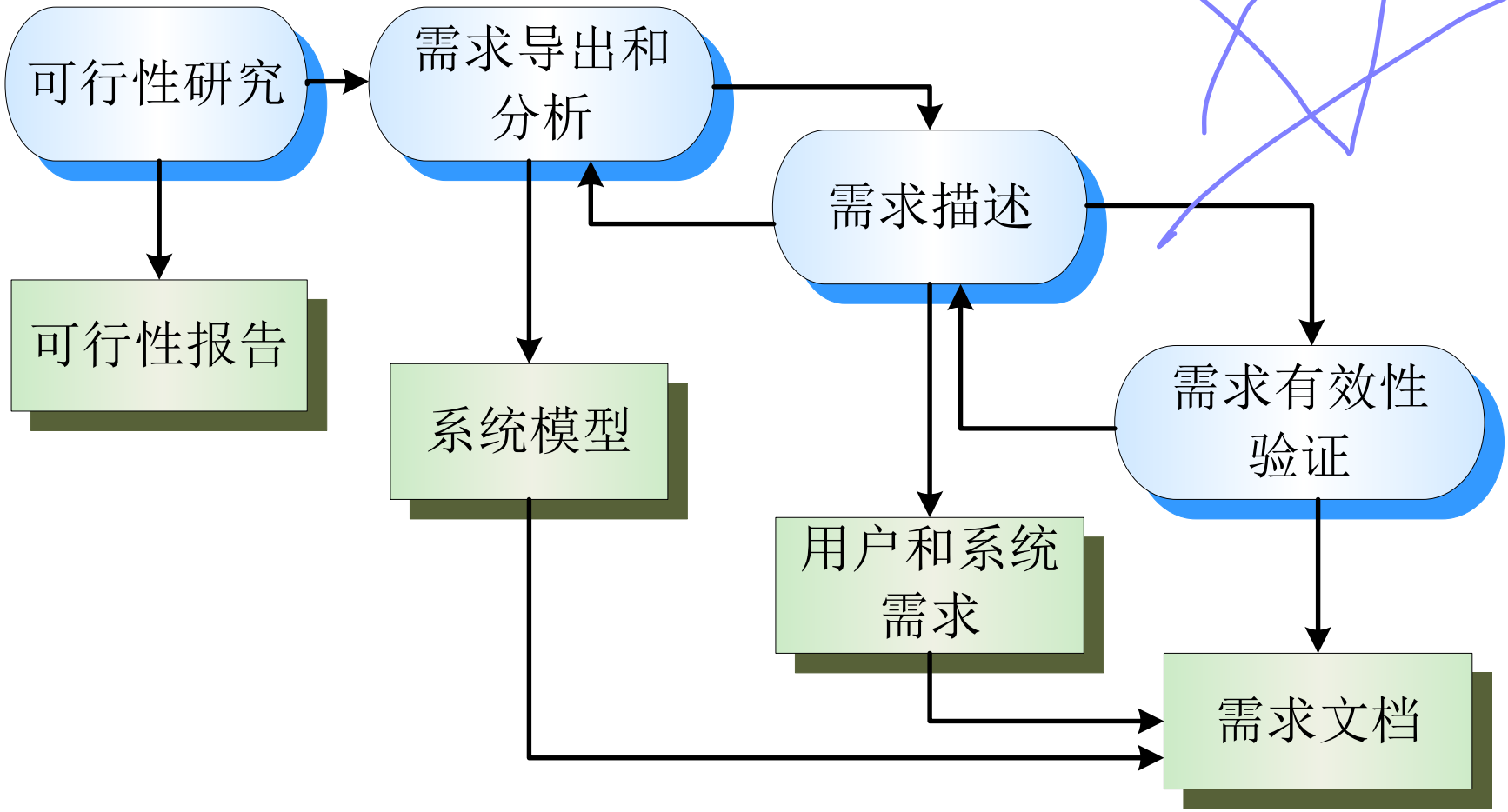
内容

- 可行性研究
- 需求导出和分析
- 需求有效性验证
- 需求管理

需求工程过程

- 需求工程过程依赖于应用领域、涉及的人员和机构
- 然而，对所有过程都存在着一些共通的活动
  - 需求导出
  - 需求分析
  - 需求验证
  - 需求管理

需求工程过程



可行性研究

需求分析的第一阶段

该阶段也需要有一定的描述，系统定义与功能的简要描述；

- 可行性研究决定一个建议的系统是否值得做
- 可行性研究过程较短，焦点集中在：
  - 系统是否符合机构的总体目标
  - 系统是否可能在现有的技术条件、预算和时间限制内完成
  - 系统能否与已经存在的其它系统集成

本机构的实际条件

很粗略的估算

比如选课系统和学籍系统，
选课系统的学生数据可以从学籍系统导入，
一方面减少开发成本，另一方面保证了数据的一致性

可行性研究的实现

- 基于信息评估(需要什么), 信息汇总和报告生成
- 向机构中人员提出的问题
  - 如果系统没有实现，机构如何应付
  - 当前处理过程的问题是什么?
  - 建议的系统会有什么帮助?
  - 什么是集成的问题?
  - 是否需要新技术、新技能?
  - 建议的系统需要提供什么?

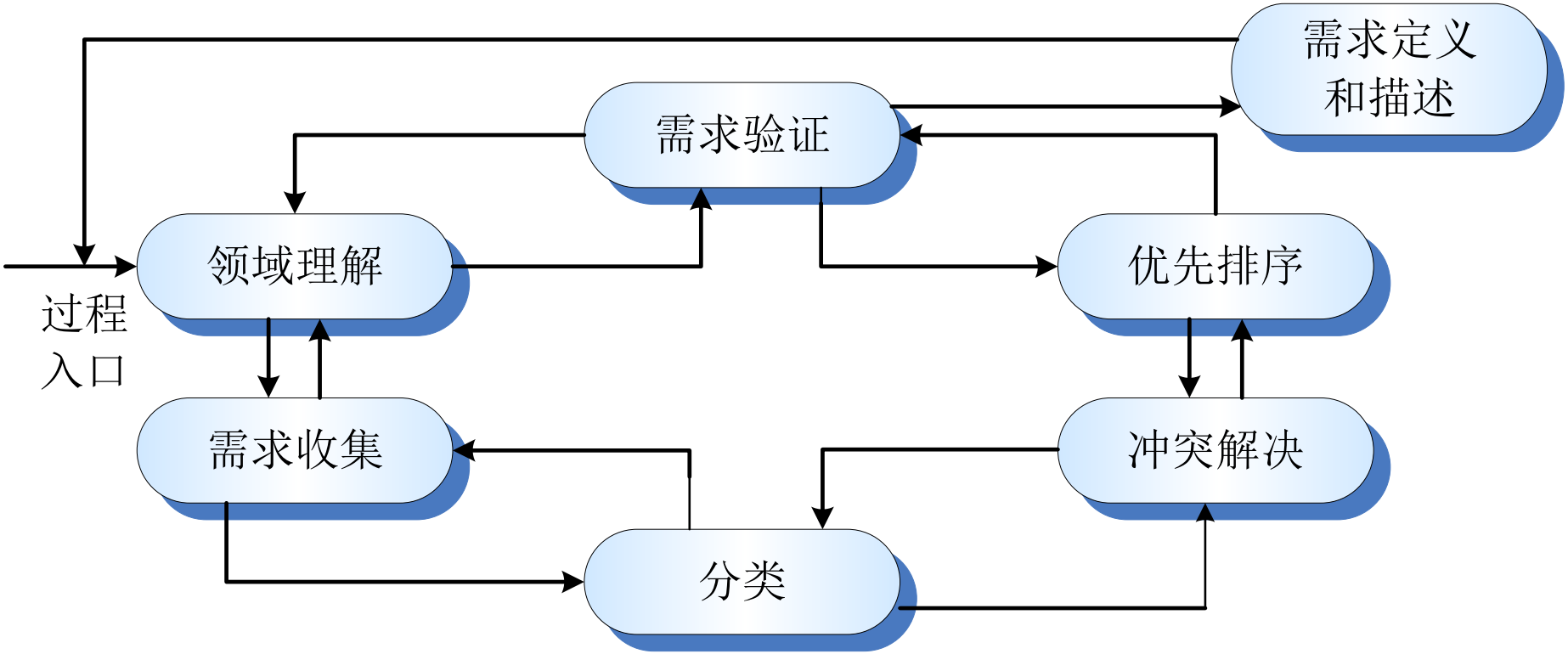
导出和分析

- 也叫需求导出或者需求发现
- 相关的技术人员和客户一起工作以发现应用领域、系统提供的服务、系统的运行限制
- 可能涉及终端用户、管理者、维护工程师、领域专家等项目相关人员。

系统分析的问题

- 项目相关人员不知道他们真正想要什么
- 项目相关人员用他们自己的语言表达需求
- 不同的项目相关人员提出的需求可能冲突
- 机构和政治上的因素可能影响系统需求
- 分析过程中需求改变。可能会出现新的项目相关人员，业务环境改变。

需求分析过程



过程活动

- 领域理解
- 需求收集
- 分类
- 冲突解决
- 优先排序
- 需求检查

系统模型

- 需求分析活动过程中产生不同的模型
- 需求分析有三种活动
  - 区分. 识别实体之间的结构关系 不断思考各个实体是否需要细分
  - 提取. 识别实体中的一般特性 对象模型（属性和方法）
  - 发散. 识别看一个问题的不同方式 比如与其他系统之间的联系
- 系统模型在第7章叙述

面向视点的导出

结构化的方法
保证需求的完备性

- 项目相关人员不同的问题视点
- 这种多视点的分析是重要的，因为分析系统需求没有单个正确的方式

银行ATM系统

- 一个银行自动柜员机的例子
- 使用一个非常简单、只提供有限银行服务的系统
- 服务包括取现金、消息传送(发消息要求服务)、订购信息和转帐

自动柜员机的视点

- 银行客户
- 其它银行的代表
- 硬件和软件维护工程师
- 市场部
- 银行管理人员和柜台工作人员
- 数据库管理员和安全工作人员
- 通信工程师
- 人事部

视点的类型

- 数据源或数据接收器
  - 视点用于生产或消费数据。分析过程包括识别所有的视点、识别产生或消费什么数据以及采取了什么处理过程。
- 表示框架
  - 视点代表特定类型的系统模型。比较这些模型以发现需求，使用单个模型的话容易出错。对实时系统特别适合。
- 服务的接收者
  - 系统外的视点，从系统接受服务。主要适合于交互系统。

外部视点

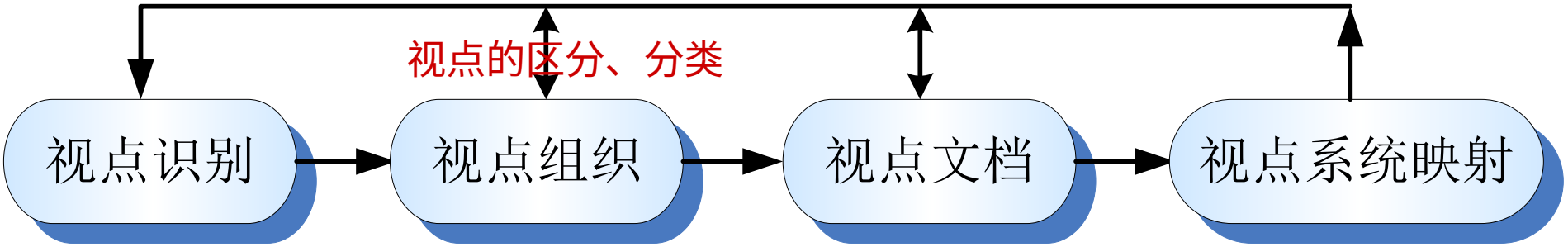
- 作为系统服务的接收者，对终端用户来说比较自然
- 组织需求导出比较自然的一种方式
- 确定一个视点是否有效，相对比较容易
- 视点和服务对于组织非功能需求非常有用

是否有效，
看系统与视点之间是否存在交互

基于方法的分析

- 广泛应用于需求分析。基于使用一个结构化方法来理解系统。
- 不同方法有不同的重点。一些是为了需求导出设计的、另一些更接近于设计方法。
- 例如，面向视点的需求定义方法 (VORD)。也说明了视点的应用。

VORD方法



视点不一定是正当用户，
也那些非法的、恶意的用户

VORD 过程模型

- 视点识别
  - 发现接收系统服务的视点，以及识别每个视点提供的服务
- 视点组织
  - 组织相关的视点形成层次结构。通用的放在较高的层次。
- 视点文档
  - 对被识别的视点和服务描述的精炼。
- 视点系统映射
  - 将分析转化为面向对象的设计

VORD模板标准格式

视点模板

名称：视点名称

属性：属性提供视点信息

事件：一组事件情景，描述了系统是如何响应视点事件的 来自视点的事件

服务：一组服务描述 功能

子视点：子视点的名称

服务模板

名称：服务的名称

原理：为什么提供该服务的理由

描述：服务的描述，用不同的符号表达

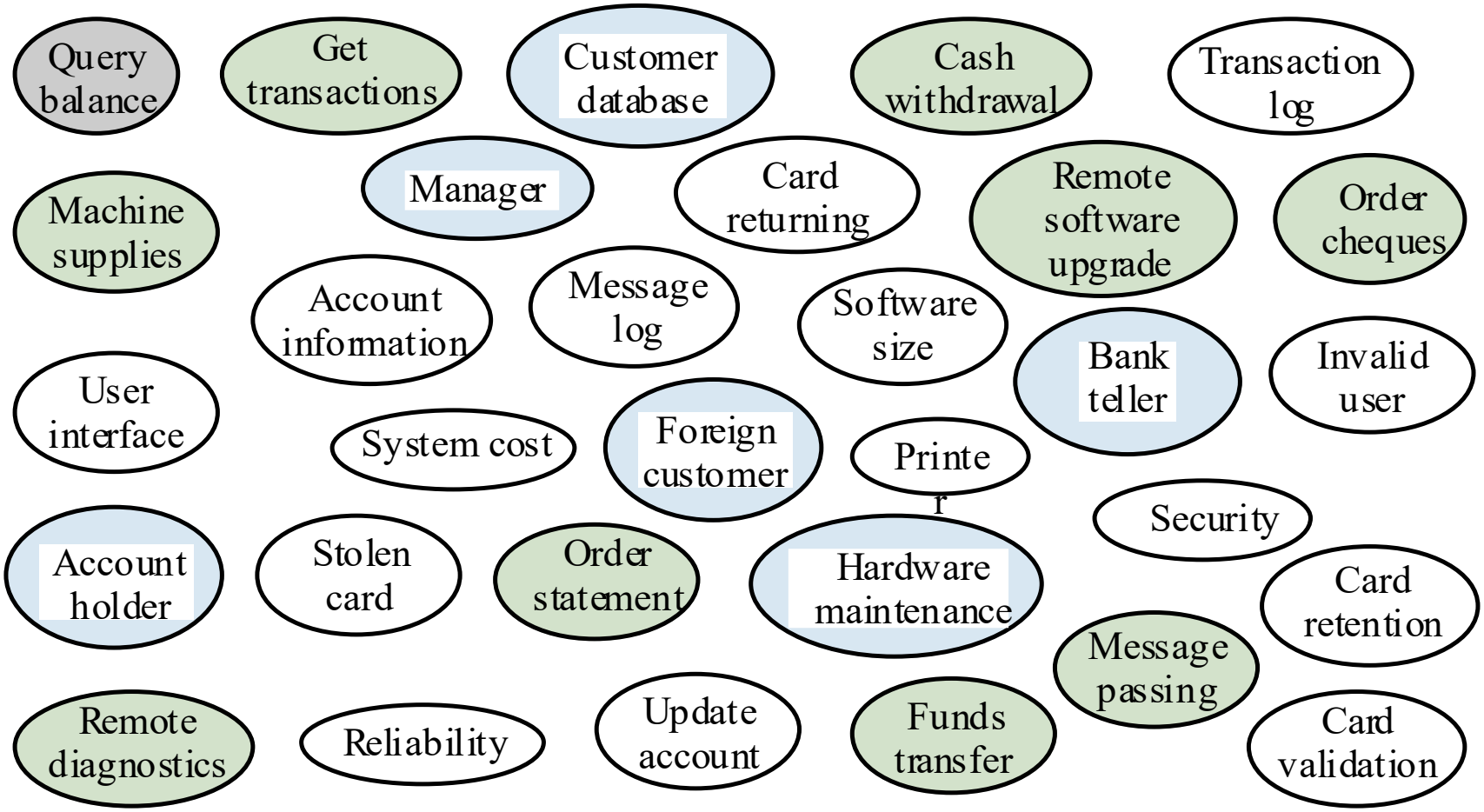
视点：接收该服务的视点列表

非功能需求：约束该服务的非功能需求

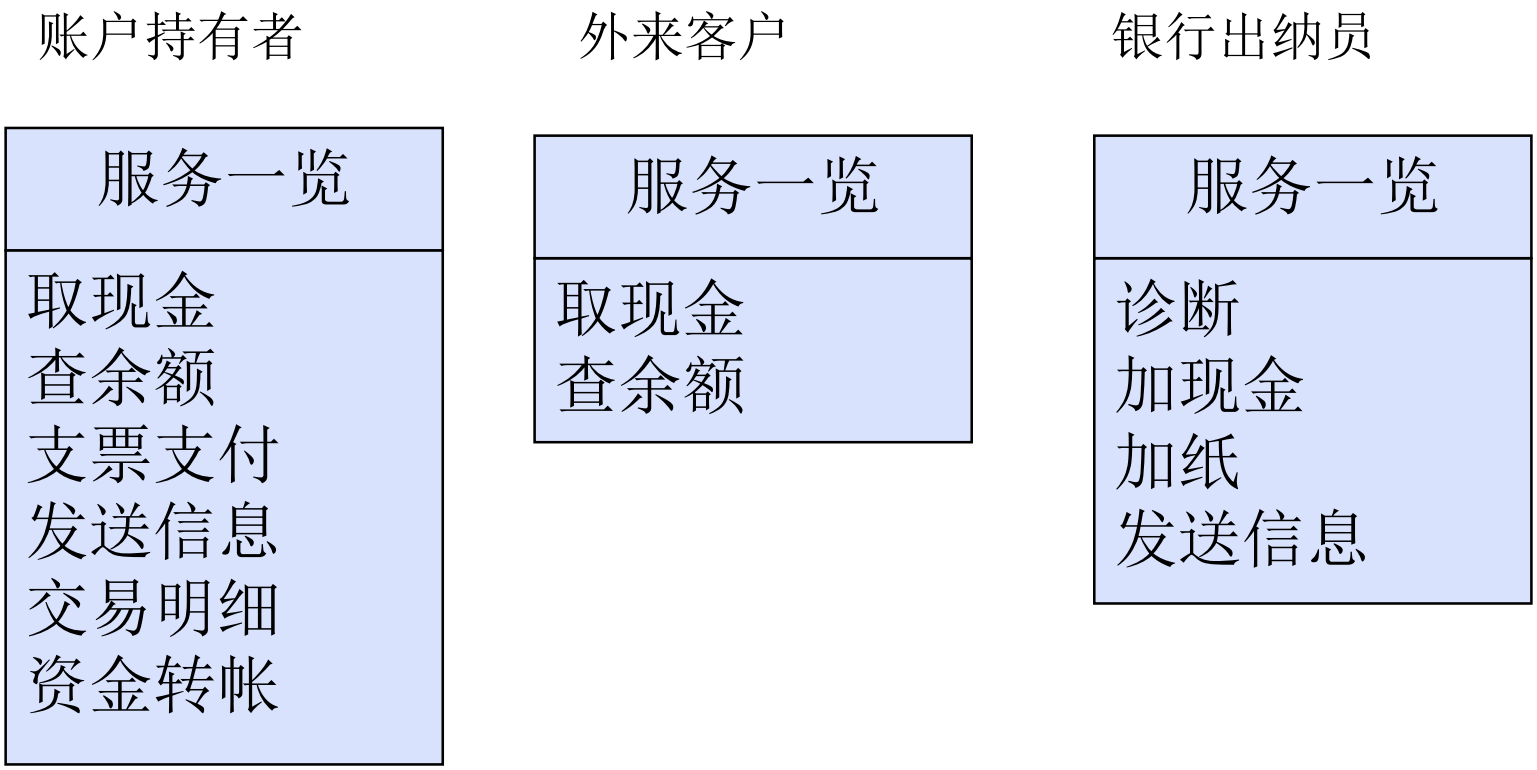
提供者：提供该服务的系统对象列表

视点识别

卡片法
比较随意，想到什么写什么；
往往写正式文档的时候会束手束脚



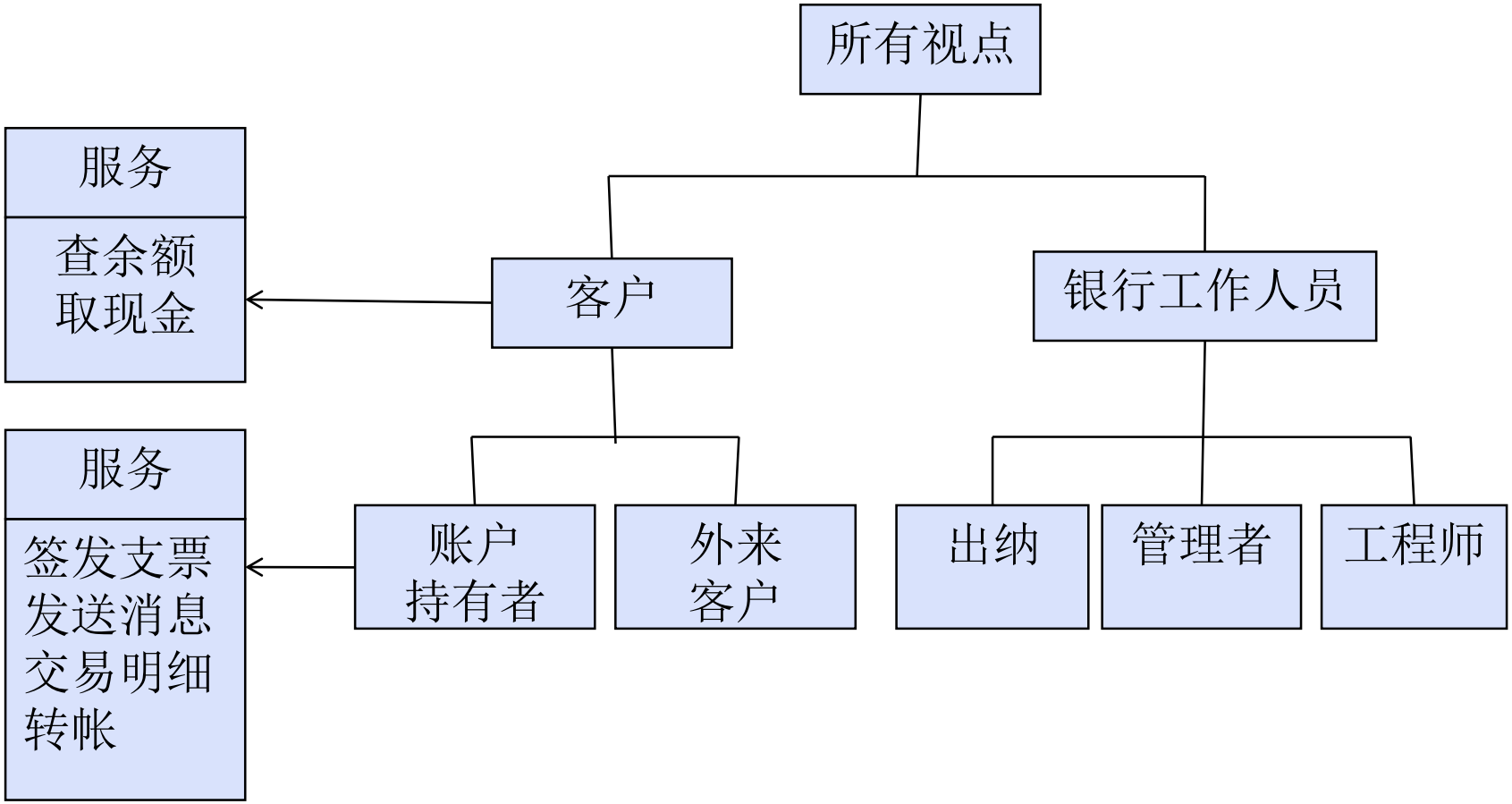
视点服务信息


账户持有者
服务一览
取现金
查余额
支票支付
发送信息
交易明细
资金转帐
外来客户
服务一览
取现金
查余额
银行出纳员
服务一览
诊断
加现金
加纸
发送信息

视点数据/控制

	控制输入	数据输入
账户持有者	开始交易 取消交易 结束交易 选择服务	卡信息 个人识别号 金额 消息

视点层次结构


所有视点
服务
查余额
取现金
客户
银行工作人员
服务
签发支票
发送消息
交易明细
转帐
账户
持有者
外来
客户
出纳
管理者
工程师

客户/取现金模板

名称：客户

属性：帐号
身份证明

事件：开始交易
选择服务
取消交易
结束交易

服务：取现金
查余额

子视点：帐户持有者
外来客户

名称：取现金

原理：为了改善客户服务、减少柜台工作

描述：用户通过按“取现金”按钮选择该服务。用户随后可以输入要求的金额。金额被确认，如果资金许可，则交付现金。

视点：客户

非功能需求：1分钟之内交付现金

提供者：*暂缺*

场景

- 场景是关于一个系统如何实际使用的描述
- 场景对需求导出是有帮助的，因为比起抽象的描述，场景更容易让人关联起来。
- 场景对于添加细节到需求描述概要中是特别有用的。

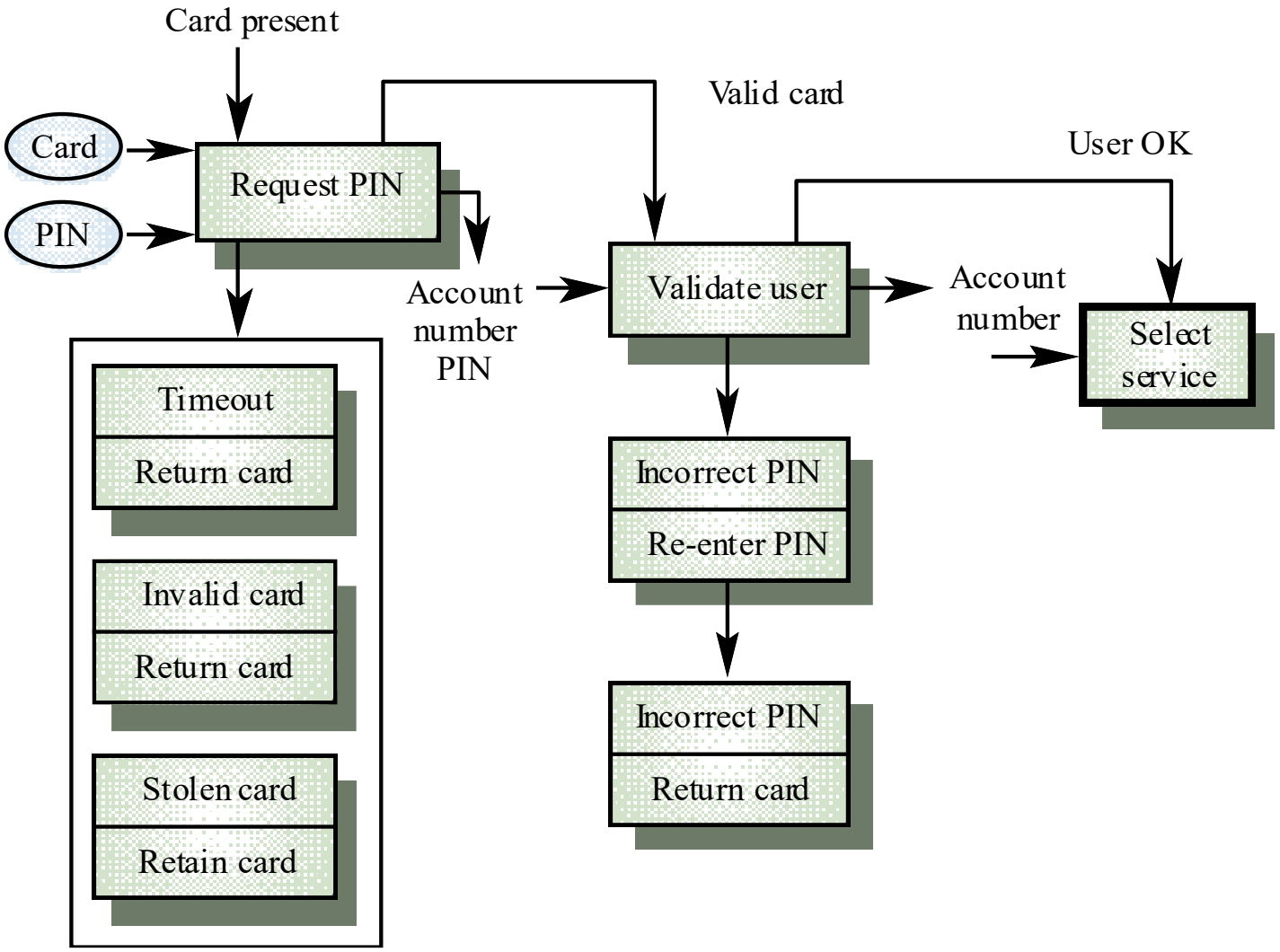
场景描述

- 场景开始时的系统状态
- 关于常规事件流的描述
- 哪里会出错以及如何处理错误
- 其它同时发生的活动的信息
- 场景完成后的系统状态

事件场景

- 事件场景可用来描述系统对特定事件如何响应，例如“开始交易”
- VORD 包括事件场景的方块图符号约定
  - 数据提供和交付
  - 控制信息
  - 例外处理
  - 下一个预期的事件

事件场景 –开始交易


Card present
Card
PIN
Request PIN
Valid card
User OK
Account number PIN
Validate user
Account number
Select service
Timeout
Return card
Invalid card
Return card
Stolen card
Retain card
Incorrect PIN
Re-enter PIN
Incorrect PIN
Return card

数据和控制分析的符号

- 椭圆. 来自视点和交付给视点的数据
- 控制信息从每个方框的顶部出入
- 数据从每个方框的右侧出来
- 异常显示在方框的底部
- 场景完成后预期的下一个事件的名称表示在一个粗线框中

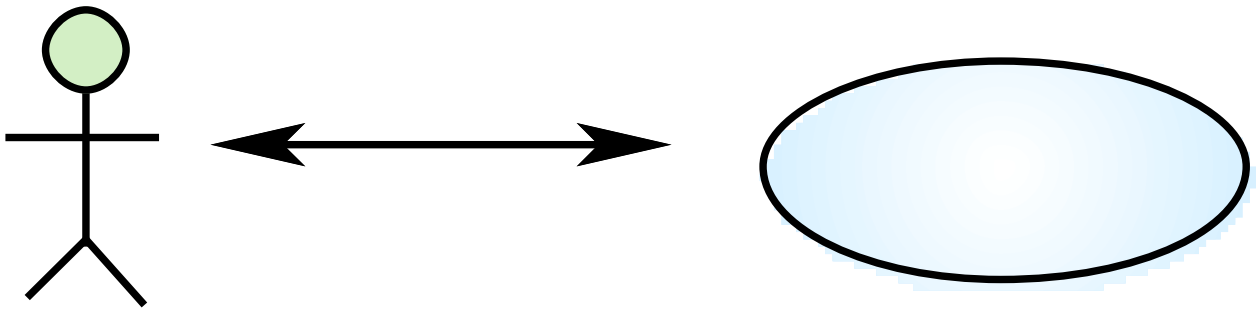
异常描述

- 多数方法不包含描述异常的功能
- 在本例中的异常是
  - 超时. 在限制时间内客户没能输入身份证号码
  - 无效卡. 卡不被认可、退回卡
  - 被盗卡. 卡是已登记的被盗卡，机器没收卡

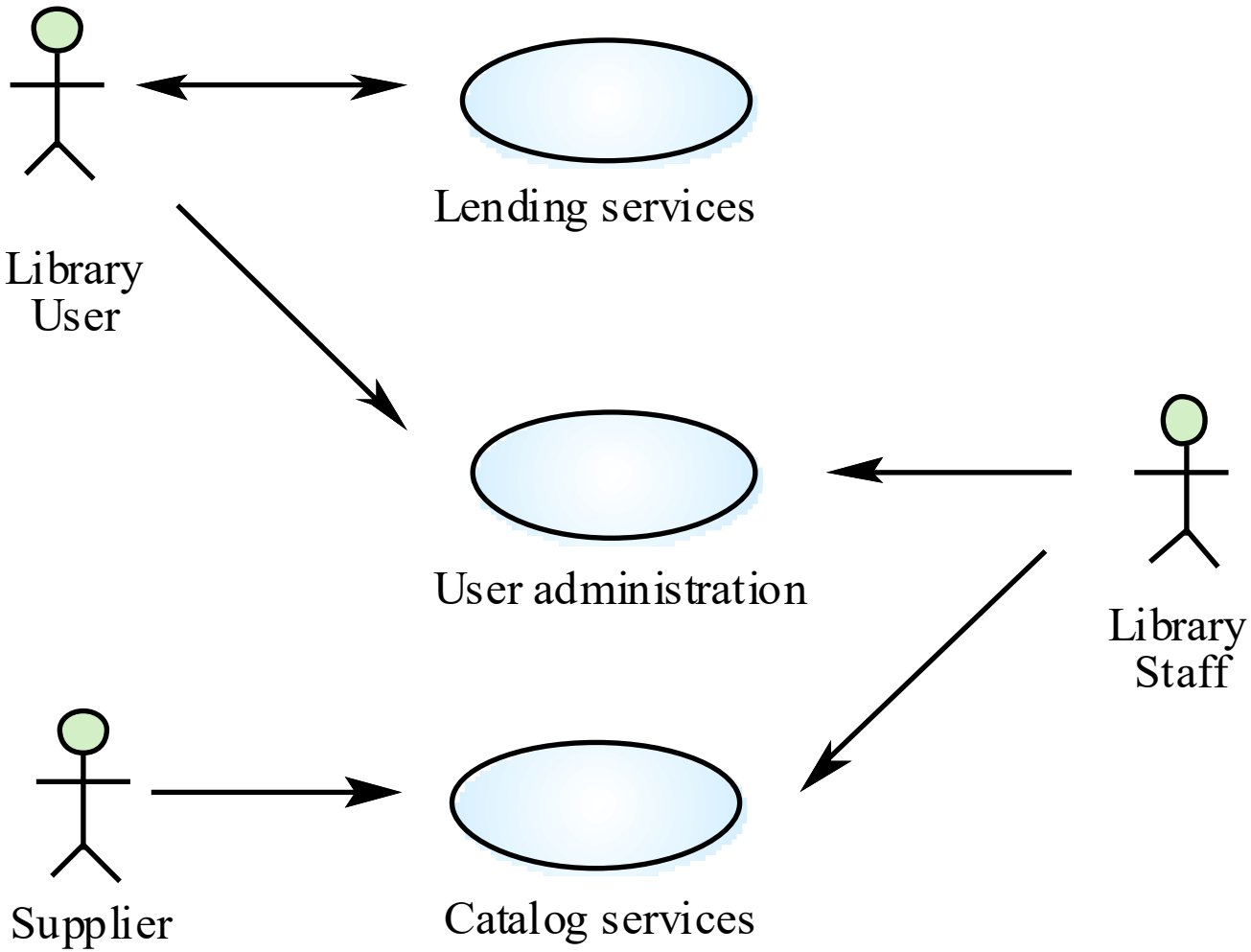
用例

- 用例是基于场景的需求导出技术、用UML标记。用例确定交互中的角色、描述交互本身。
- 一组用例应描述系统所有可能的交互
- 通过表示系统中的事件处理序列，使用序列图向用例中添加细节。

出借用例

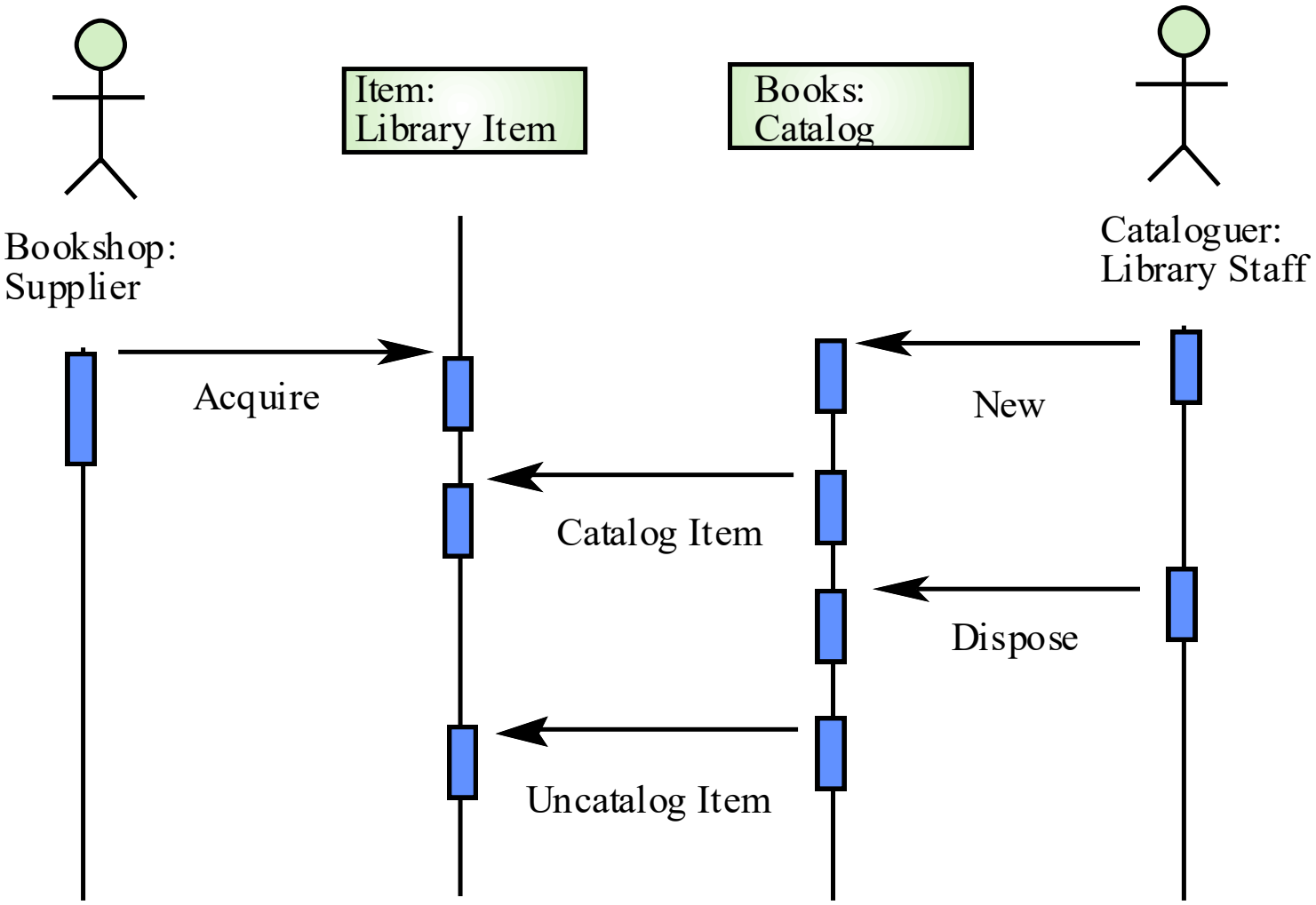


图书馆用例


Lending services
Library
User
User administration
Library
Staff
Supplier
Catalog services

目录管理的序列图

大部分模块是不需要序列图的，
一般是需要强调过程细节的模块才需要



社会和机构的因素

- 软件系统应用于社会和机构中，所以社会和机构的因素会影响系统需求、甚至占支配地位。
- 社会和机构的因素不是单个视点、而是对所有视点产生影响。
- 好的分析人员应该对这些因素感觉敏锐，但是目前还没有一种系统化的方法来处理这样的分析。

举例

- 考虑这样一个系统，允许高层管理人员直接访问信息，不通过中层管理人员。
  - 管理地位. 高层管理者可能觉得他们地位很高不愿意用键盘。这可能限制了系统接口的可用类型。
  - 管理职责. 管理者可能没有连续的时间来学会使用系统。
  - 机构的阻力. 系统使用后可能会多余的中层管理人员可能会故意提供误导的或者不完整的信息以让系统最终失败

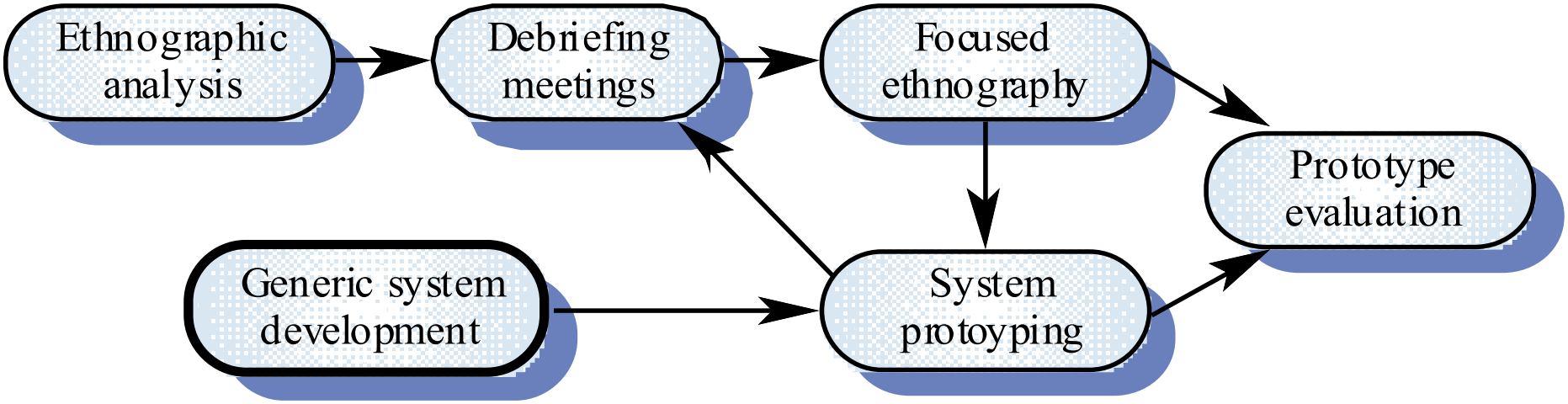
行为学

- 社会学家花了大量的时间观察和分析人们是怎样真正地工作的。
- 解释和说明他们的工作比较困难
- 难于观察社会和机构的因素
- 人种学研究表明，真实的工作通常比假设的简单系统模型要更丰富更复杂。

关注行为学

- 在空中交通管制系统项目中提出
- 将行为学与原型建立结合起来
- 原型开发出现行为学相关的未知问题
- 行为学的问题：行为学研究以往的实践活动，有些过去的基础现在已经不存在了。

行为学和原型开发


Ethnographic analysis
Debriefing meetings
Focused ethnography
Prototype evaluation
Generic system development
System protoyping

行为学的范围

- 需求源自于人们真实的工作，而不是所建议的
- 需求源自于协作以及了解其他人的工作

需求有效性验证

- 证明系统中定义的需求是客户真正想要的
- 需求错误的代价是很高的，所以有效性验证非常重要
  - 交付后修改一个需求错误比起修改一个实现上的错误，其代价是高至后者的100倍。

需求检查

- 有效性： 系统是否提供了最适合于客户的功能
- 精确性： 需求是否是模棱两可的?
- 一致性： 有没有冲突的地方?
- 完整性： 是否包括了客户需要的所有的功能?
- 现实性： 在一定的预算和技术条件下需求是否能够实现
- 可验证性： 需求是否可以检验?

需求有效性验证技术

- 需求评审
  - 对需求做系统性的手工分析
- 原型开发
  - 用一个可执行的系统模型来检查需求。在第8章详述
- 测试案例生成
  - 检查需求的易测性
- 自动一致性分析
  - 检查结构化需求描述的一致性

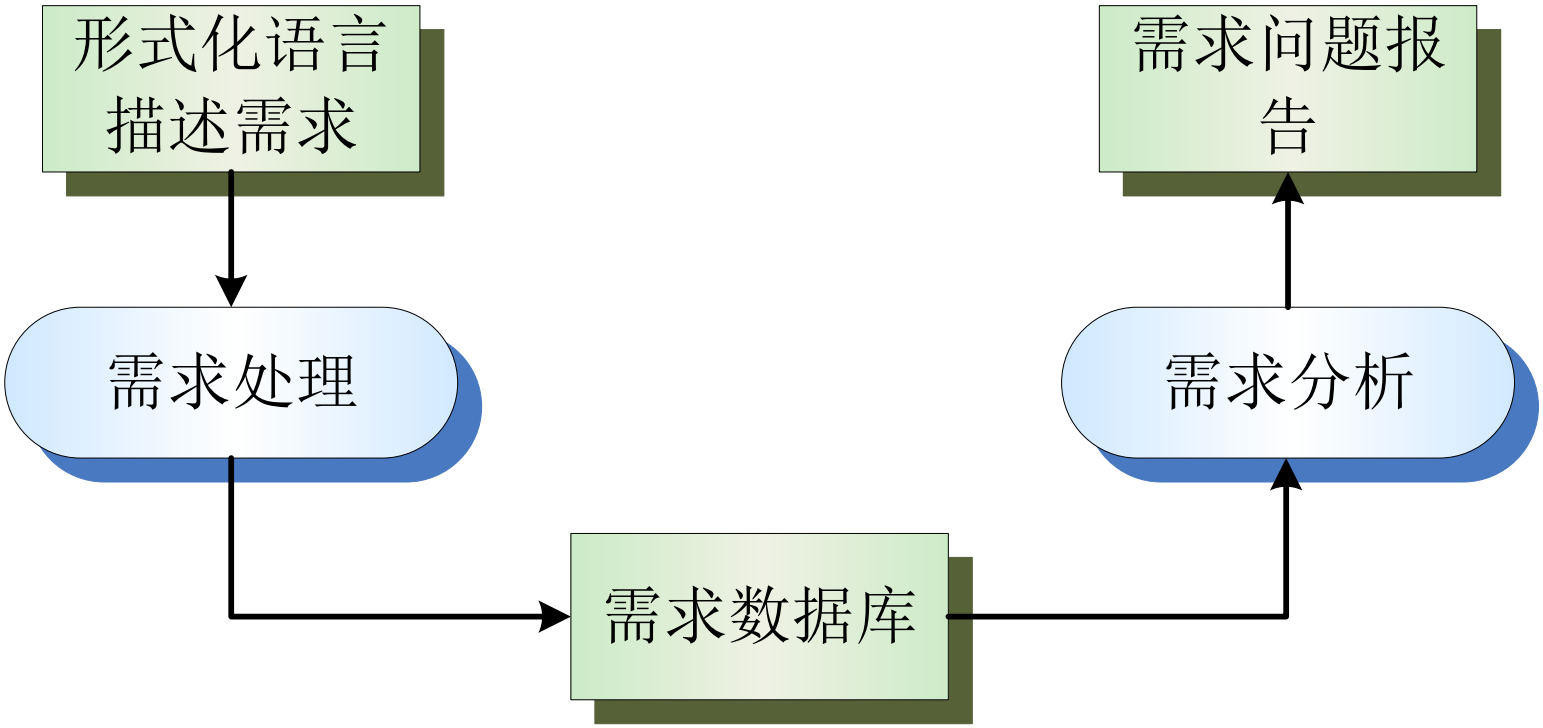
需求评审

- 举行经常性的评审会、阐明需求定义
- 客户方和承包商方的工作人员都应参加评审
- 需求可以是正式的（具有完整的文档）也可以是非正式的。开发者、客户和使用者之间良好的沟通可以在早期解决问题。

评审的检查

- 可检验性。需求实际上能否测试?
- 可理解性。需求是否被正确理解?
- 可追溯性。需求的来源是否清晰的说明?
- 适应性。 在对别的需求没有大的影响的情况下、需求能否改变?

自动一致性检查



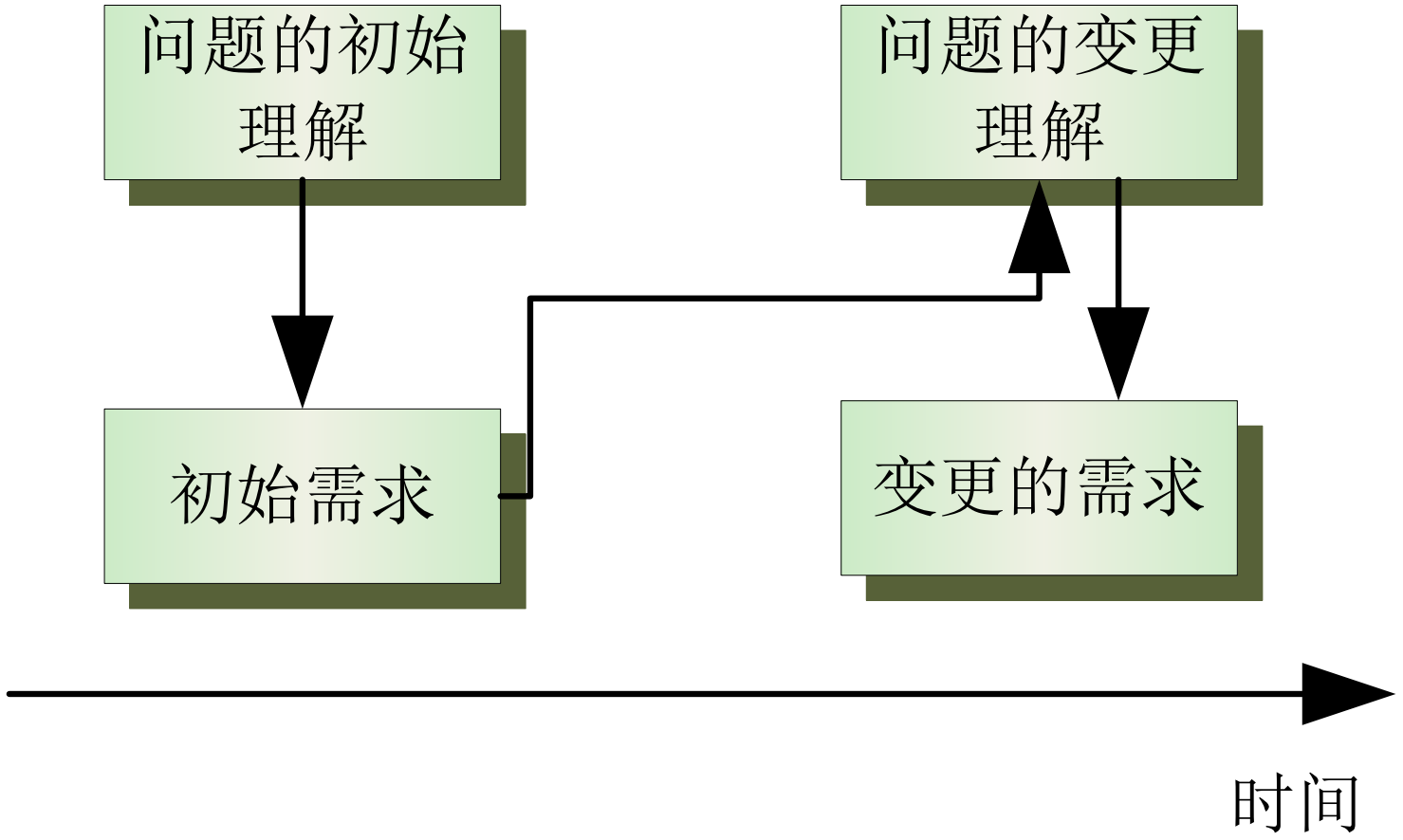
需求管理

- 需求管理是在需求工程过程和系统开发过程中管理需求变更的过程
- 需求不可避免是不完整、不一致的
  - 当业务需求变化、系统理解更深入时，都会出现新的需求
  - 不同的视点有不同的需求，这些经常是矛盾的。

需求变更

- 在开发过程中，源自不同视点的需求的优先级是变化的。
- 系统客户从商务角度描述的需求会跟终端用户的需求冲突。
- 开发过程中系统的商务和技术环境变化

需求进化



持久和易变的需求

- 持久的需求. 稳定的需求来自客户核心的活动。例如一个医院总是有医生、护士。可以从领域模型得到。
- 易变的需求. 开发或者系统使用中变化的需求。如医院，来自卫生保健政策的需求。

需求的分类

- 易变的需求
  - 由于环境的改变导致需求的变化
- 浮现的需求
  - 对系统有了更深的理解后，新的需求浮现出来
- 引发的需求
  - 由于计算机系统的引入所带来的需求
- 兼容性需求
  - 需求依赖于机构中其它系统或业务过程

需求管理规划

- 在需求工程过程中，必须计划以下内容:
  - 需求识别
    - » 需求如何被唯一的标识
  - 变更管理过程
    - » 分析需求变更的过程
  - 可追溯策略
    - » 需求之间以及需求和设计之间的关系加以记录和维护
  - CASE工具支持
    - » 需要工具来帮助管理需求变更

可追溯性

- 可追溯性是指需求、需求的来源和设计之间的关系
- 来源可追溯性
  - 将需求和提出需求的项目相关人员之间建立连接
- 需求可追溯性
  - 在独立的需求之间建立连接
- 设计可追溯性
  - 在需求和设计之间建立连接

可追溯性矩阵

Req. id	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2
1.1		U	R					
1.2			U			R		U
1.3	R			R				
2.1			R		U			U
2.2								U
2.3		R		U				
3.1								R
3.2							R	

U：代表行上的需求使用了列上需求的服务
R：代表行列需求之间存在其它较弱的关系

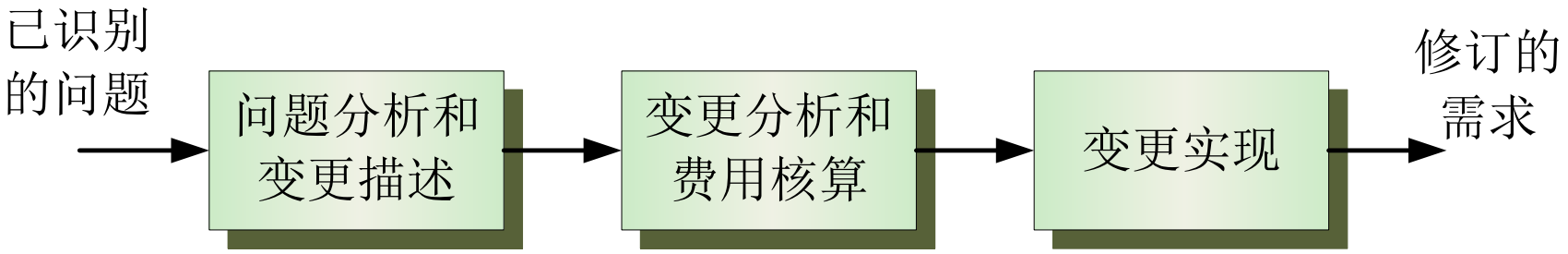
CASE工具支持

- 需求存储
  - 需求的存储应该是安全的、有效管理的
- 变更管理
  - 变更管理的过程由有效的工具来支持会变得简单
- 可追溯性管理
  - 借助工具找到相关的需求

需求变更管理

- 对所有建议的需求变更都应管理
- 基本阶段
  - 问题分析. 讨论需求的问题、建议变更
  - 变更分析和成本计算. 评估变更对其它需求的影响
  - 变更实现. 修改需求文档和其它文档以反映变更

需求变更管理



要点

- 需求工程包括一个可行性研究、需求导出和分析、需求描述、需求有效性验证及需求管理
- 需求分析是一个包括领域了解、需求收集、分类、组织、优先排序和有效性验证的重复过程
- 不同的项目相关人员对系统有不同的需求

要点

- 社会和机构的因素对系统需求具有强大的影响
- 需求有效性验证是检查需求的有效性、一致性、完备性、现实性和可检验性的过程
- 业务上的变化不可避免地导致需求变更
- 需求管理过程包括规划和变更管理