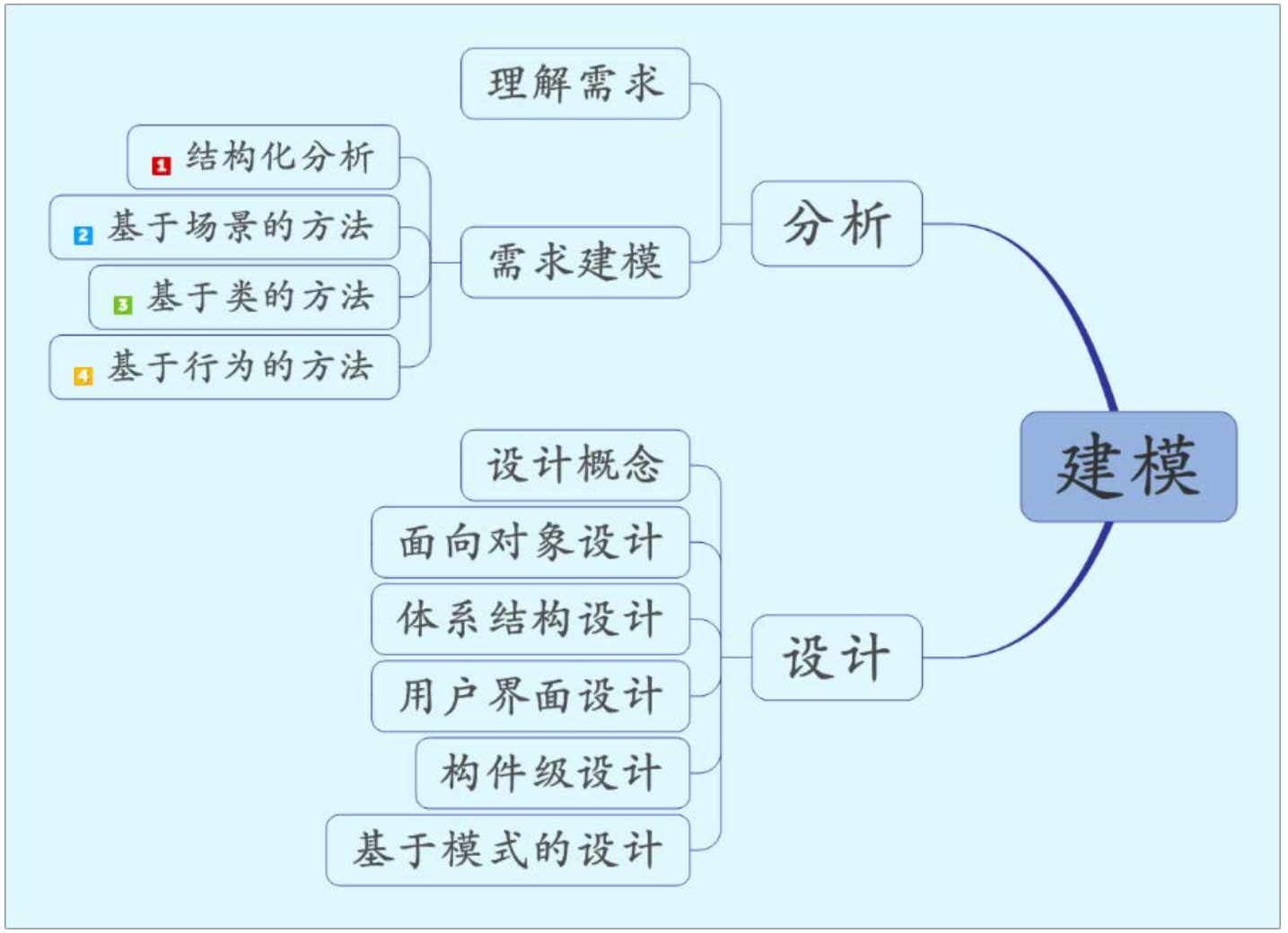
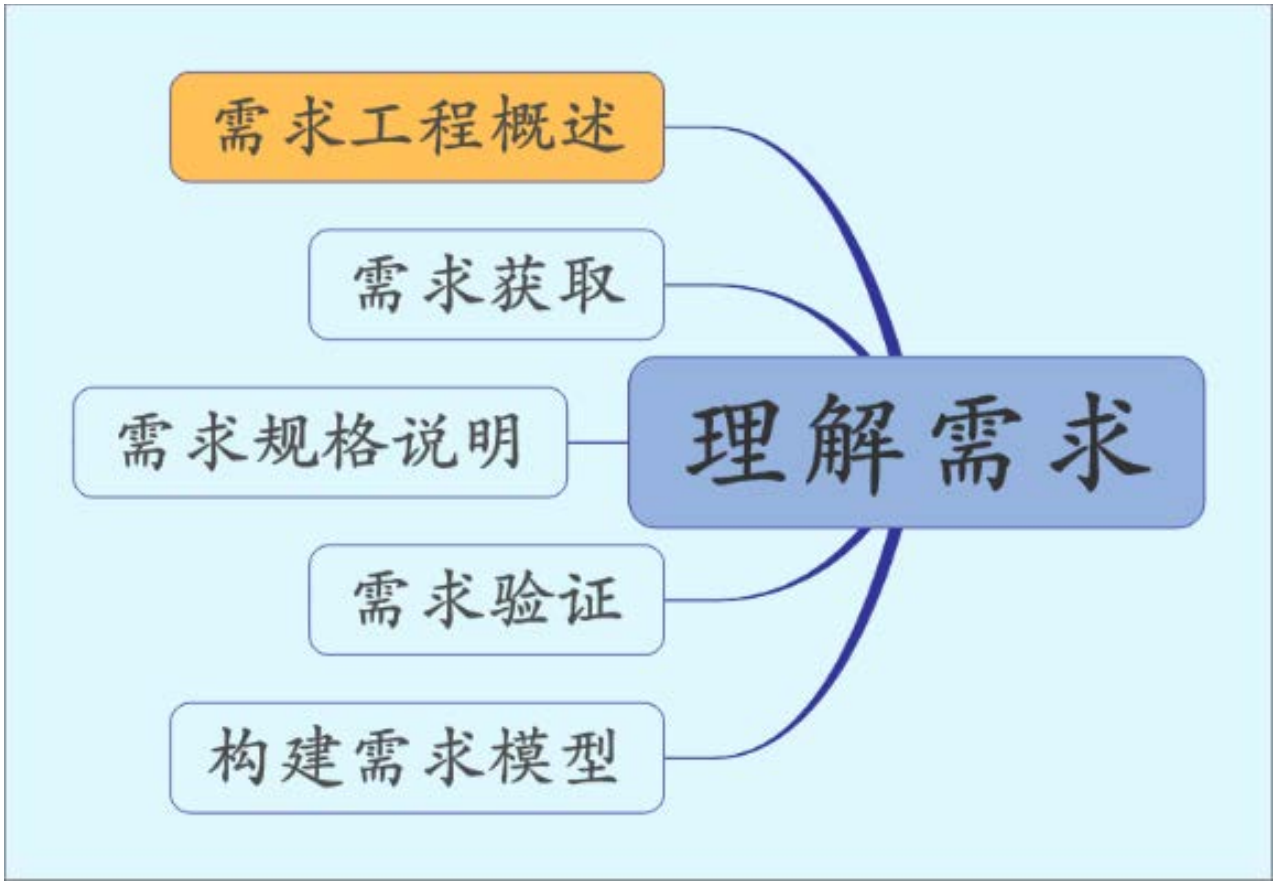
第二部分　建模


建模
分析
理解需求
需求建模
结构化分析
基于场景的方法
基于类的方法
基于行为的方法
设计
设计概念
面向对象设计
体系结构设计
用户界面设计
构件级设计
基于模式的设计

理解需求
(Understanding Requirements)

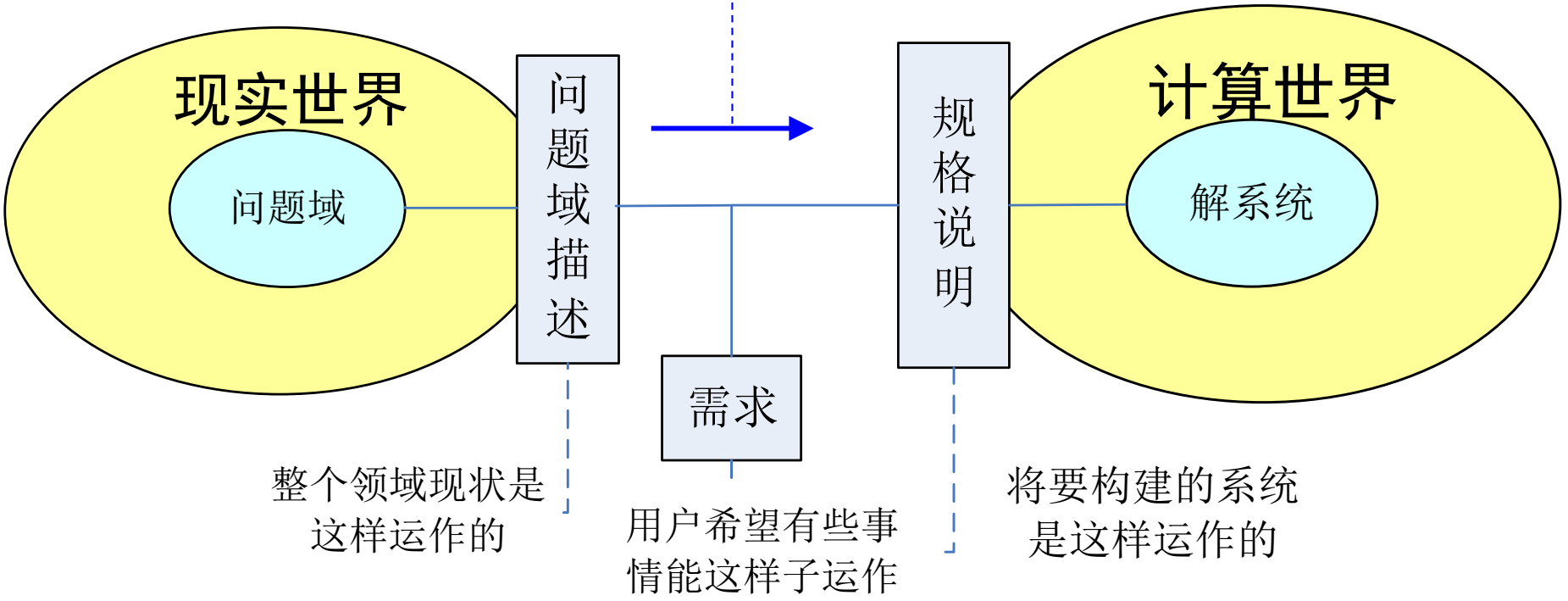
outline



Frederick Brooks

- “开发软件系统最为困难的部分就是准确说明开发什么。最为困难的概念性工作便是编写出详细技术需求，这包括所有面向用户、面向机器和其它软件系统的接口。同时这也是一旦做错，将最终会给系统带来极大损害的部分，并且以后再对它进行修改也极为困难。”

需求工程的目标


现实世界
问题域
问题域描述
规格说明
计算世界
解系统
需求
整个领域现状是这样运作的
用户希望有些事情能这样子运作
将要构建的系统是这样运作的

需求概述

- “目标系统必须做什么？”
- 确定目标系统必须具备的功能
- 需求分析是软件生存期的一个重要阶段，是软件开发项目得以成功的基础。
- 需求的重要性
  - 软件开发的基础和前提
  - 最终目标软件系统验收的标准
  - 避免或者尽早剔除早期的错误

需求的重要性

- Standish Group调查了350多家的8000多个软件项目
  - 31%的软件项目在完成之前就取消了
  - 在大公司中, 只有9% 的项目按时在预算内交付
  - 在小公司中，满足这些标准的有16%
- 结论：开发人员难以在预算内交付正确的系统

需求的重要性

- **Standish 请求被调查者解释其项目失败的原因**

  1. incomplete requirements (13.1%)

  2. lack of user involvement (12.4%)

  3. lack of resources (10.6%)

  4. unrealistic expectations (9.9%)

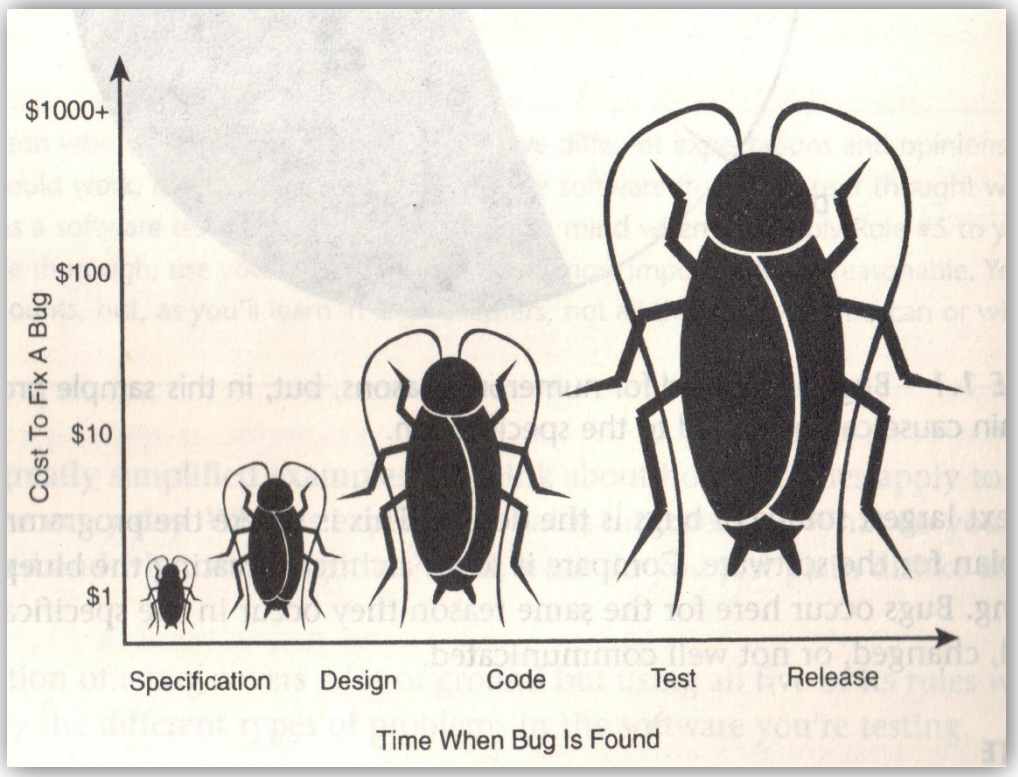
  5. lack of executive support (9.3%)

  6. changing requirements and specifications (8.7%)

  7. lack of planning (8.1%)

  8. system no longer needed (7.5%)

需求的重要性



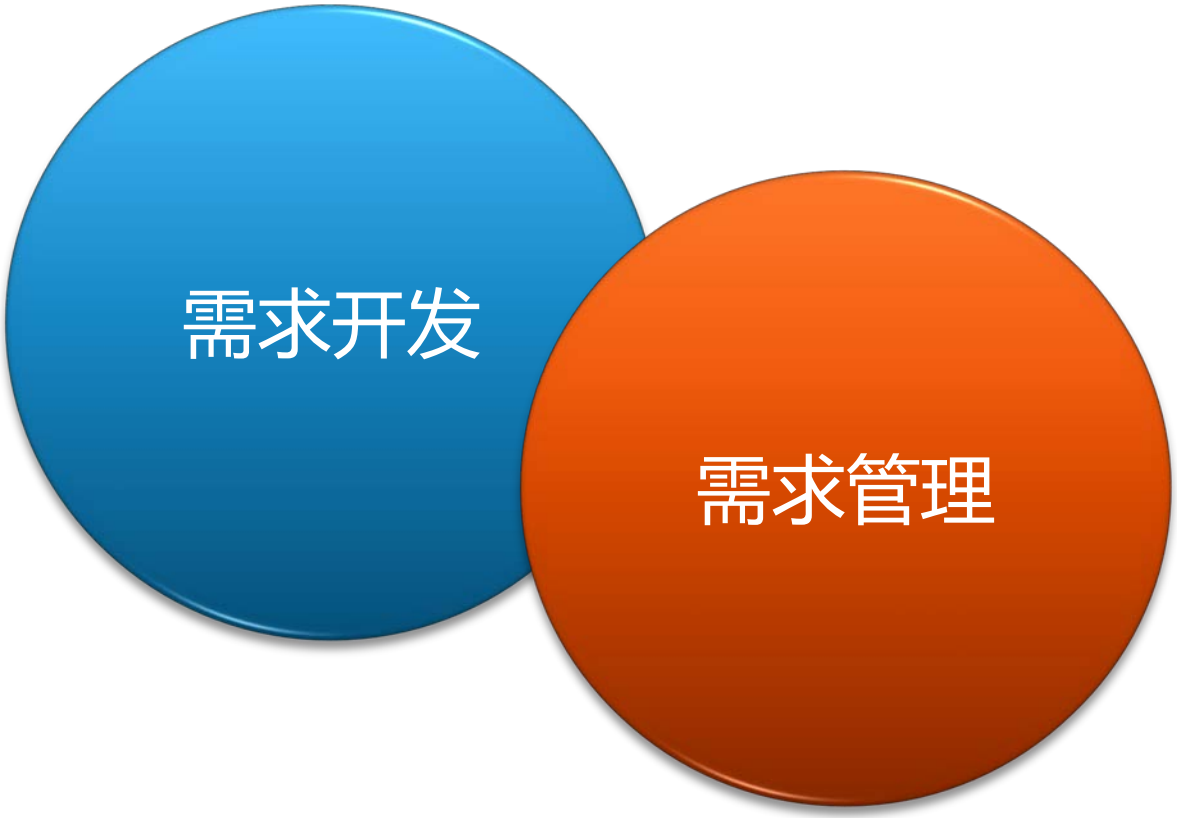
The 1-10-100 Rule

- 软件生命周期中，一个错误发现得越晚，修复错误的费用越高。

什么是需求工程

- 软件需求作为软件生存周期的第一个阶段，其重要性越来越突出，到20世纪80年代中期，逐步形成了软件工程的子领域——需求工程。
- 需求工程：对系统应该提供的服务和所受到的约束进行理解、分析、建立文档、检验的过程。

需求工程



需求开发

工作基础

- 需求获取
- 需求分析
- 编写需求规格说明
- 需求确认

需求管理

需求的两个层次

软件需求的两个层次：

- 用户需求(user requirements)
  - 一般性(general)
- 系统需求(system requirements)
  - 详细的(detailed)

用户需求

- 执行实际工作的用户对系统所能完成的具体任务的期望，描述了系统能够帮助用户做些什么。
- 通常通过用户访谈、调查，对用户使用的场景进行整理，从而建立用户角度的需求。
- 用户需求是从用户角度描述的系统功能需求和非功能需求，通常只涉及系统的外部行为，而不涉及系统的内部特性。
- 通常使用自然语言和图来描述。

系统需求

- 系统需求是关于软件系统功能和运行约束的更详细的描述。
- 定义了系统中需要实现的功能，描述了开发人员需要实现什么 。
- 合同的一部分
- **一个用户需求可以扩展为多个系统需求**

需求实例：病人信息管理系统

用户需求

1. 系统每月生成管理报告，显示每个门诊在该月开出的药物的价格。

系统需求

1. 系统在每月的最后一个工作日生成一份关于所开药物、药物的价格以及开药门诊的清单。
2. 系统在每月最后一个工作日的17:30以后自动生成该管理报告。
3. 系统应为每个门诊生成一份报告，列出药物的名称、所有的处方、所开药物的剂量、所开药物的总价。
4. 只有在访问控制列表中的授权用户才能够访问管理报告。

需求的两种类型

软件需求可以分为：

- 功能需求 (functional requirement)
- 非功能需求 (non-functional requirement)

功能需求

- 描述系统应该提供的功能或服务
- 功能需求应当具备完整性和一致性
  - Complete：用户的所有需求
  - Consistent：涉众间的需求无冲突
- 实际情况：困难
  - 对于大型系统，理解功能需求时容易犯错误或遗漏
  - 涉众间存在不同的、以及不一致的需求

功能需求实例

1. 医生能够创建病人的病历，编辑系统中的有关信息，浏览病人的病历等。
2. 系统定期检测病人的记录，如果病人有一段时间没有来复诊，则发出警告信息。
3. 病人应能够检索其预约清单。
4. 系统应能够生成每天每个门诊预约看病的病人清单。
5. 医院每位使用系统的职员，其身份应由其8位员工号确定。

……

非功能需求

- 系统的特性和约束
  - 性能(performance)、可靠性(reliability)、安全性(security)、可用性(availability)等
  - 与其它系统的接口、系统运行的软硬件环境等
- 通常比功能需求更关键
  - 用户设法绕过某些不能真正满足其需求的功能
  - 不能满足性能需求的系统将无法使用

非功能需求

性能	可靠性	安全性	可用性	接口
• 执行速度 • 响应时间 • 吞吐量 • 数据接收或发送的时间间隔	• 平均失效间隔时间 • 失效后重新启动系统允许的最大时间	• 控制对系统的访问 • 将每个用户的数据和其他用户的数据隔离开	• 用户理解并使用系统的难易程度 • 每类用户需要的培训类型	• 输入是来自一个还是多个其他系统 • 输出是否传到一个或多个其他系统

非功能需求分类

- 产品需求(Product requirements)
  - 详细描述软件产品的特性
  - 性能、占用内存空间、可靠性、安全性、可用性等
- 组织需求(Organizational requirements)
  - 组织的策略和过程
  - 开发过程需求、使用过程需求、环境需求等
- 外部需求(External requirements)
  - 外部因素引起的需求
  - 法律需求、互操作需求等

非功能需求实例

产品需求

1. 系统在工作时间（周一至周五，8:00至18:30）对所有诊室可用。
2. 系统的维护时间应安排在周一至周五的21:00至午夜，周日的8:00至12:00
3. 不同层级的医生能够快速学会使用系统。
4. 系统应在2秒内响应用户的单个查询请求。
5. 如果系统的响应超过2秒，则通过进度指示告知用户正在进行数据查询处理。
6. 系统在每天的宕机时间不应超过5秒。
7. 系统应使用基于角色的访问控制技术，使得不同角色的用户能够访问不同的信息。

非功能需求实例

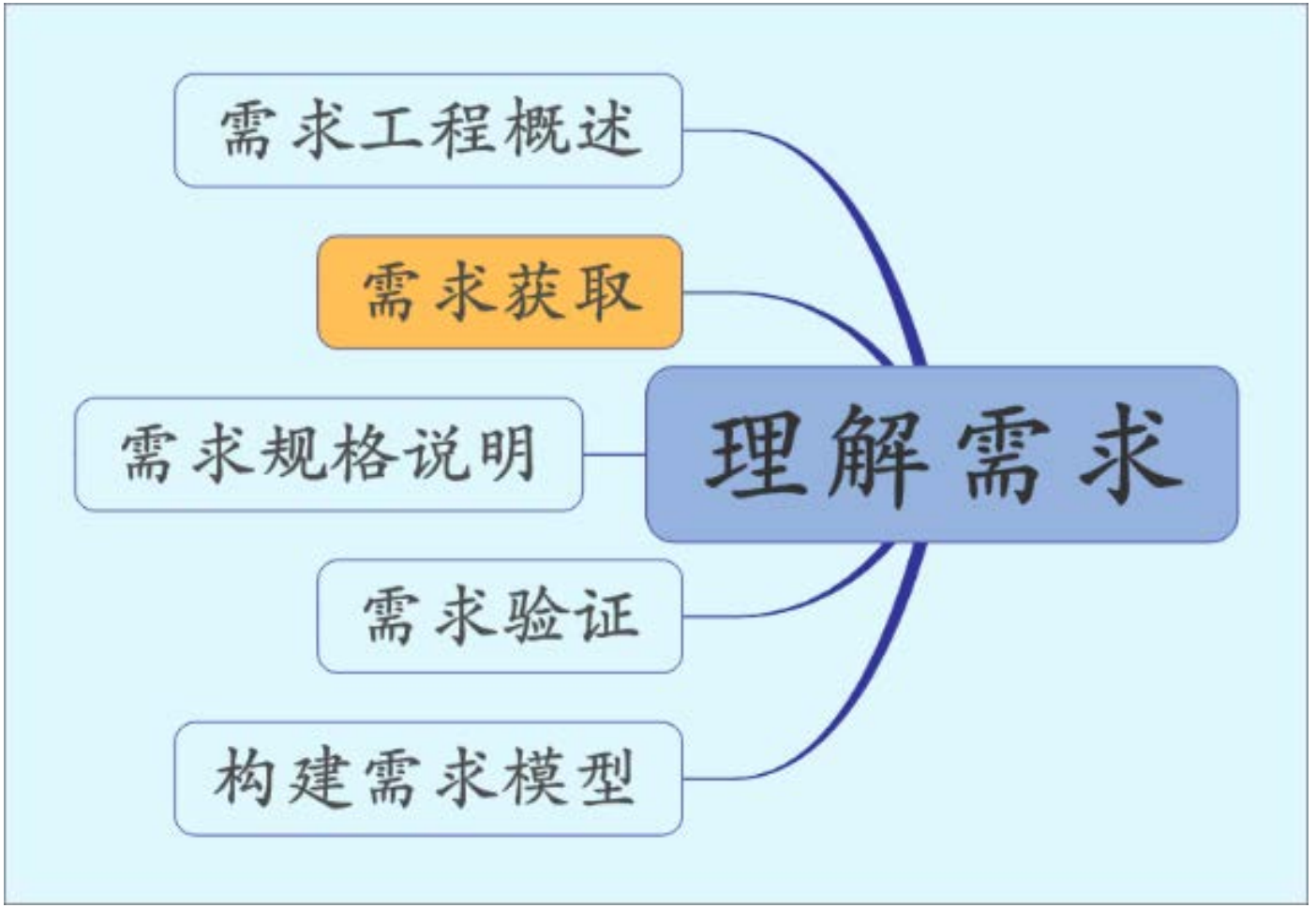
组织需求

1. 系统通过读取医疗管理机构颁发的身份卡对用户进行身份认证。
2. 系统运行于医院数据中心的Linux服务器上，服务器的最大可使用内存为32GB。

外部需求

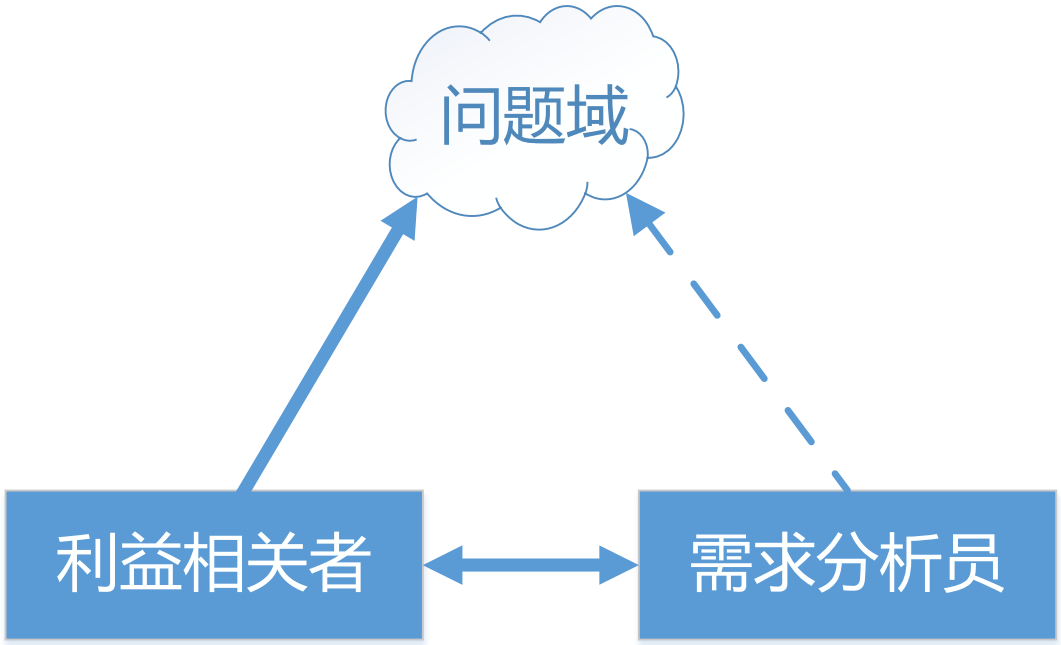
1. 系统应确保只有经过授权的医生和管理人员才能访问病人的病历信息。
2. 系统应与国家药物信息管理系统交互，通过接口读取其中关于药物特性和价格等信息。
3. 系统应与国家病历系统交换信息。
4. 系统应与已有的病人预约系统交换预约数据。
5. 系统应调用已有开药系统中的开药服务实现开药的功能。
6. 系统应使用国家医疗机构单点登录认证系统。

outline



概述

- 需求分析员与利益相关者一起发现系统需求
  - Stakeholders：最终用户、管理人员、工程师、领域专家等



困难

- 利益相关者可能不知道自己真正想要的是什么，可能会提出不切实际的需求
- 利益相关者通常以他们自己的方式表达需求，分析员由于缺乏领域知识，导致不理解需求
- 利益相关者通常用不同方式表示需求，导致需求间的**冲突**
  - 分析员必须发觉所有需求的来源，以及需求的共性和冲突
- 在分析过程中，需求会发生变更
  - 出现新的利益相关者，提出新的需求
  - 业务环境发生变化

解决冲突

- 请求利益相关者对需求进行优先级划分
  - 绝对要满足的需求（必需的）
  - 非常值得要的但并非必需的需求（值得要的）
  - 可要可不要的需求（可选的）

解决冲突

- 信用卡记账系统
  - 应能够列出最近的费用，及还款截止日期（必需的）
  - 应能够按照购买类型区分费用（值得要的）
  - 用黑色打印贷方账目，用红色打印借方账目（可选的）

需求获取技术

- 获取的信息来源
  - 文档、利益相关者和相似系统的规格说明
- 交流的对象：利益相关者
- 需求获取技术
  - 访谈 (Interviews)
  - 场景 (Scenarios)
  - 用例 (Use cases)
  - 原型 (prototypes)

病人信息管理系统的利益相关者

- 病人
- 医生
- 护士
- 医疗接待员：管理病人预约
- IT人员：安装和维护系统
- 医疗伦理管理人员：确保系统满足对病人治疗的伦理指南
- 医疗管理人员：从系统获取管理信息
- 医疗记录人员：负责系统信息的维护和保存

访谈

- 与利益相关者的正式或非正式访谈
- 访谈的两种类型
  - 封闭式访谈(closed interviews)：事先准备好问题
  - 开放式访谈(open interviews)：没有事先准备，分析人员和利益相关者共同探讨一些问题
- 有效的访谈
  - 虚心，避免对于需求有先入为主的想法，乐于倾听利益相关者的意见
  - 通过使用需求建议、原型系统等鼓励被访谈者积极讨论

场景分析技术

- 用户将来如何使用系统的**现实例子**
- 一个场景应包括
  - 场景初始情形的描述
  - 场景中常规事件流的描述
  - 场景中可能出错及如何解决问题的描述
  - 同时发生的其它活动的信息
  - 场景结束时系统状态的描述
- 场景分析步骤
  - 和利益相关者交流，确定场景
  - 获取包含在场景中的详细信息

场景实例：上传照片

初始情形

- 用户有多个数字照片要上传到系统中。这些照片保存在电脑上。
- 用户已成功登录系统

常规

- 用户选择上传照片，并且选择保存照片所属的项目名称
- 用户输入与每个上传的照片相关联的关键字
- 上传完成后，系统自动通知项目监管者，请他们检查新上传的照片

可能出现的问题

- 所选择的项目没有关联的监管者，自动通知系统管理员，请他们任命一个项目监管者
- 具有相同名字的照片已被同一个用户上传，询问用户是否希望：重新上传同名的照片，重命名照片或者取消上传

其他活动

- 监管者可以登录到系统中，并可以在照片上传时批准其共享

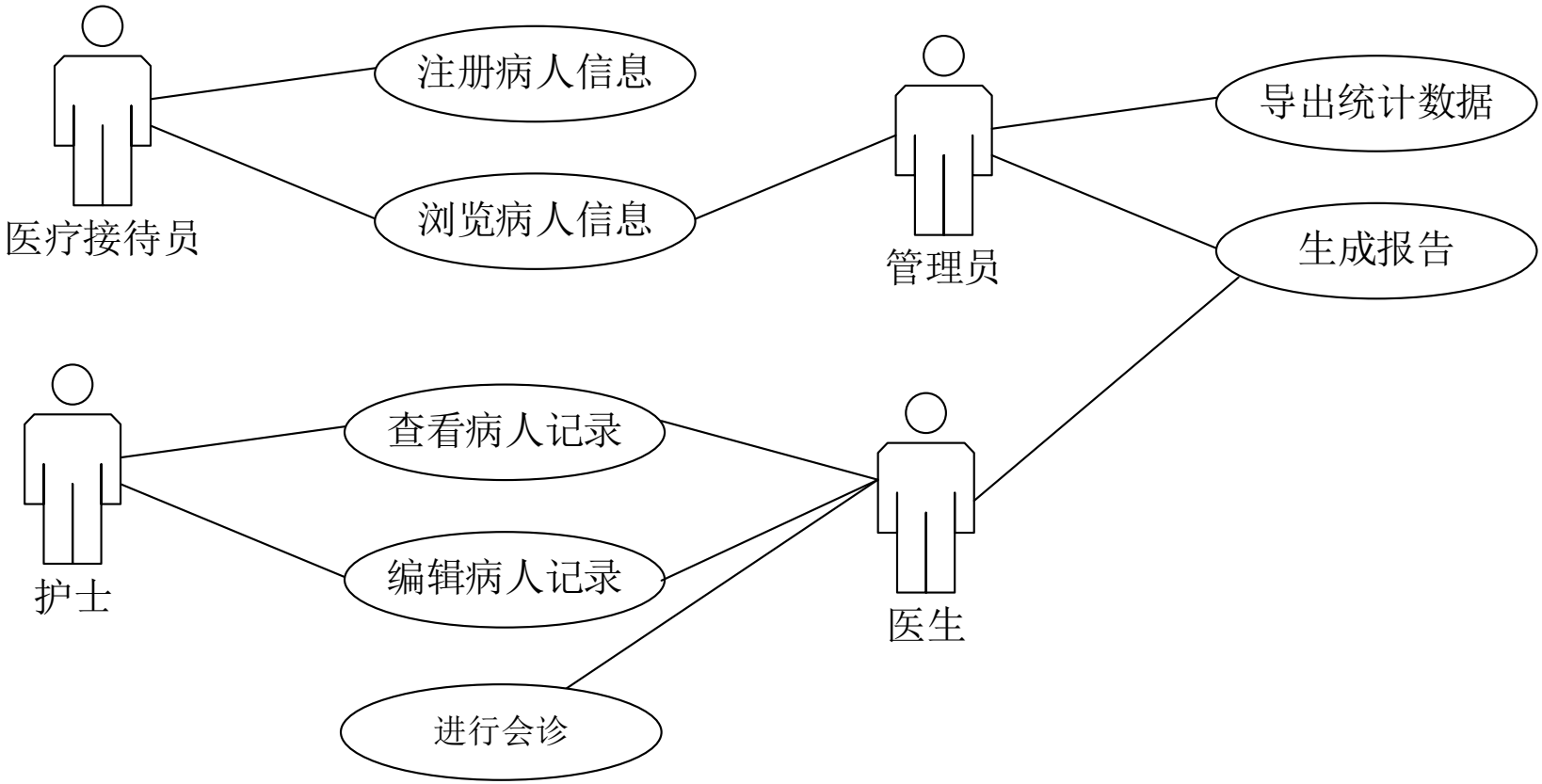
结束时系统状态

- 用户登录成功。所选择的照片已经上传并且状态是“等待批准”。照片对于监管者和上传照片的用户可见

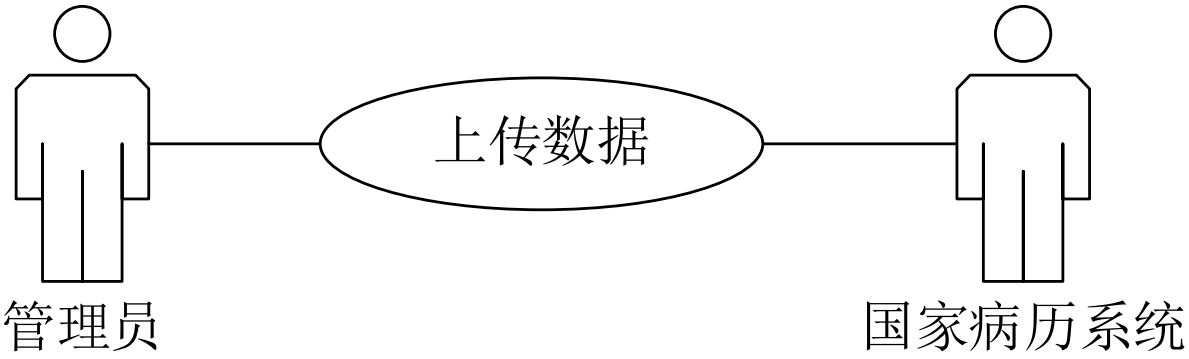
用例

- 面向对象方法
  - UML用例图
- 需求获取中广泛使用的一种方法
- 用例描述的是一种交互（interaction）
  - 利益相关者和系统
  - 系统和系统
- 用例图是对交互的一种简单概述，可以使用文本、表格或者顺序图对其补充说明
- 场景和用例没有区别
  - 每个用例就是一个场景

用例实例

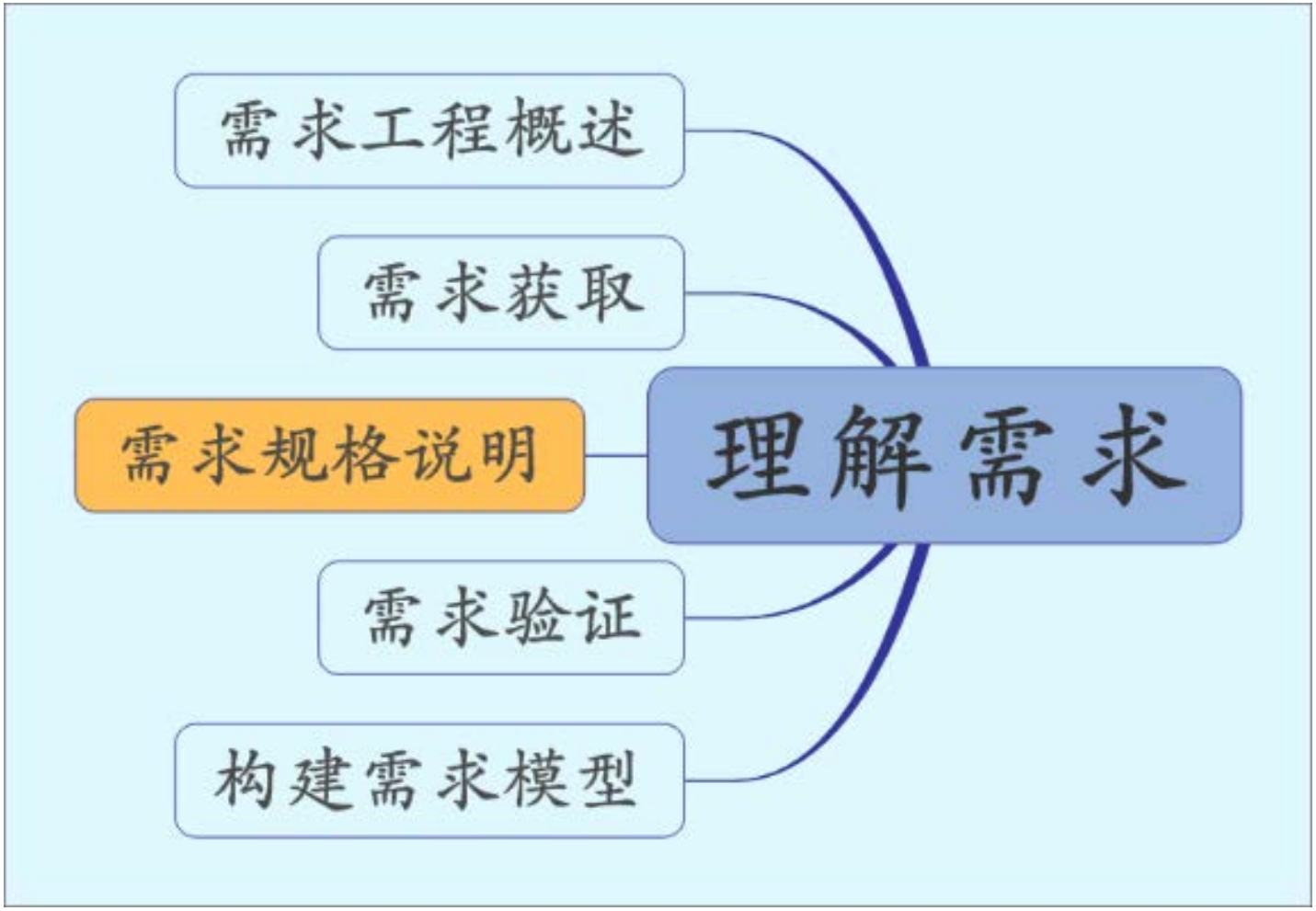

医疗接待员
注册病人信息
浏览病人信息
管理员
导出统计数据
生成报告
护士
查看病人记录
编辑病人记录
进行会诊
医生

用例实例



病人信息管理系统: 上传数据	
外部行为者	管理员，国家病历系统
描述	管理员将病人信息管理系统中的数据上传至国家病历系统的数据库中，上传的数据为更新的个人信息（住址、电话号码等），以及病人诊断和治疗的总结信息。
数据	病人的个人信息，治疗信息
触发因素	管理员发起命令
响应结果	国家病历系统已更新的确认信息
备注	管理员必须有访问病人信息和国家病历系统的安全权限

outline



需求规格说明

- 将用户需求和系统需求文档化的过程
- 软件需求规格说明（software requirements specification, SRS）
  - 用户需求不包含技术信息
    - 不应使用软件术语、结构化符号或者形式化符号
    - 使用自然语言，加上简单的图形和表格
  - 系统需求包含技术信息，是系统设计的基础
    - 使用**自然语言、图形模型**、数学模型
- SRS依赖于所开发系统的类型和开发过程模型
  - 外包软件：SRS要详细、清晰
  - 极限编程：增量式获取用户需求，用户故事

自然语言规格说明

- 优点：表达能力强、直观、具有普适性
- 缺点：潜在的模糊性和二义性
- 书写指南
  - 使用一种标准格式，建议尽量用一两句话书写需求
  - 以一致的方式使用语言：必须满足的需求用“必须”，期望满足的需求用“应该”
  - 使用强调性的文本（粗体、颜色）来突出需求中的关键部分
  - 尽量避免使用软件工程专业术语、缩写等，比如体系结构

3.2 系统必须每10分钟测量一次血糖，如果需要的话就供应一次胰岛素。

结构化规格说明

- 标准的方式而不是自由文本书写需求
- 使用**模板**来刻画系统需求

胰岛素泵/控制软件/SRS/3.3.2

功能	计算胰岛素剂量：安全的血糖水平
描述	当前测量的血糖水平在安全区间3~7个单位时，计算要供给的胰岛素剂量
输入	当前血糖读数（r2），前两个读数（r0和r1）
来源	当前读数来自传感器，其他读数来自存储
输出	CompDose——要供给的胰岛素剂量
目的地	主控制环
动作	如果血糖水平稳定或在下降，或虽然在升高但上升率在下降，那么CompDose为0；如果血糖水平在升高并且上升率也在增长，那么CompDose通过将当前血糖水平与前一血糖水平之差除以4并四舍五入来计算；如果结果为0，那么将CompDose设为可以供应的最小的剂量
要求	有前两个读数，这样可以计算血糖水平的变化率
前置条件	胰岛素存储存有至少一个所允许的最大单剂量胰岛素
后置条件	r0被r1替代，而r1被r2替代
副作用	无

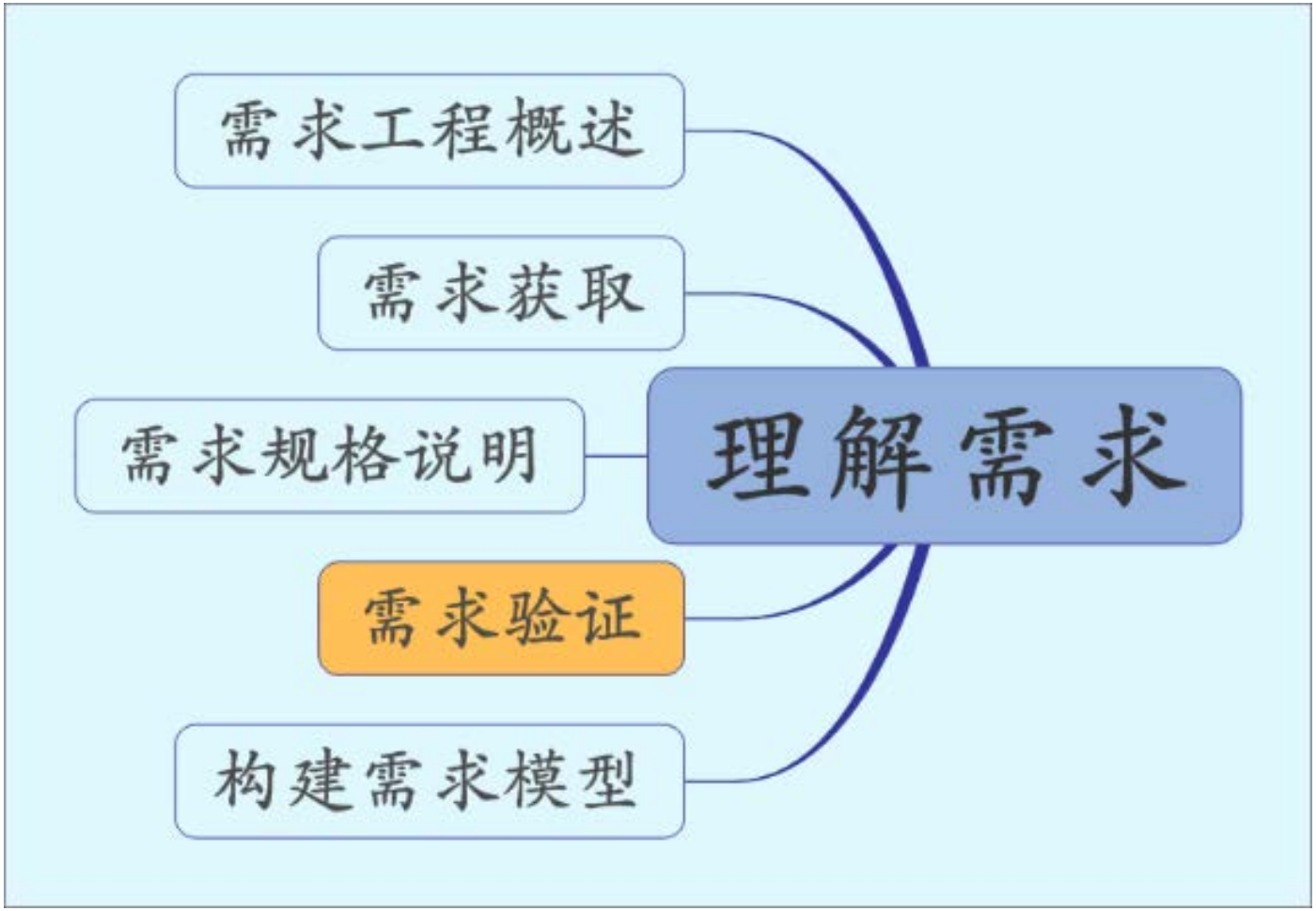
结构化规格说明

条件	动作
血糖水平在下降(r2<r1)	CompDose=0
血糖水平稳定(r2=r1)	CompDose=0
血糖水平在升高并且上升率在下降((r2-r1)<(r1-r0))	CompDose=0
血糖水平在升高并且上升率稳定或在增加 r2>r1 & ((r2-r1)>=(r1-r0))	CompDose=round((r2-r1)/4) 如果四舍五入后为0，那么CompDose=MinimumDose

SRS结构

章	描述
序言	文档的预期读者、版本信息等
引言	简略的描述系统的功能，系统和其它系统的关系，描述系统如何满足组织的战略目标等。
术语表	文档的技术词汇
用户需求定义	为用户提供的服务，非功能需求，使用自然语言、图或者其它用户理解的符号，定义产品和过程标准。
系统架构	所预计的系统体系结构的高层概览，子模块组织等。
系统需求规格说明	更详细地描述系统功能和非功能需求，定义与其它系统的接口。
系统模型	图形化的系统模型，展示系统的构件之间的关系，以及系统和环境之间的关系。对象模型、数据流模型等。
附录	和当前开发的应用相关的详细的、具体的信息，包括硬件需求，数据库描述等。
索引	字母顺序索引、图索引、函数索引等。

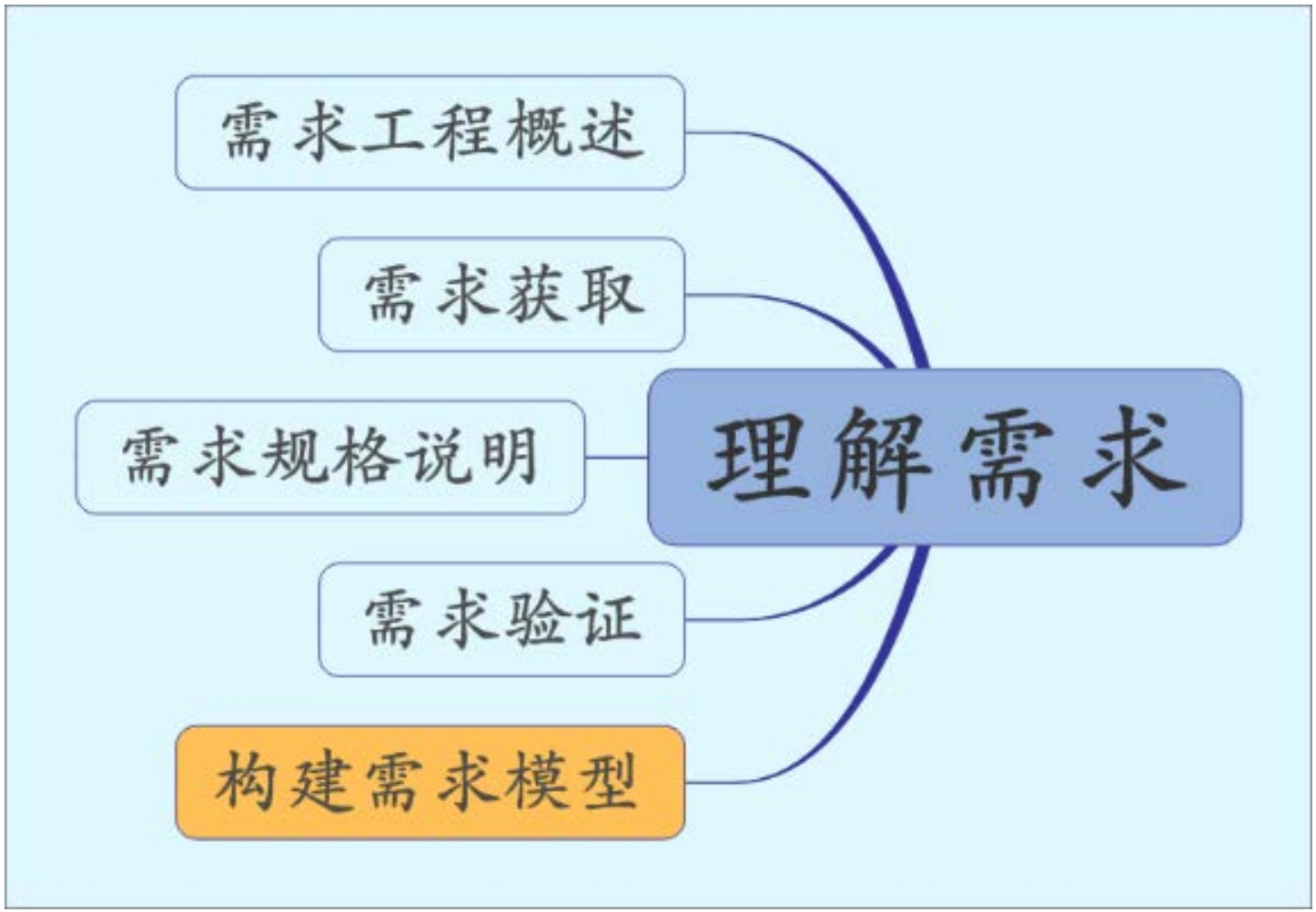
outline



需求验证

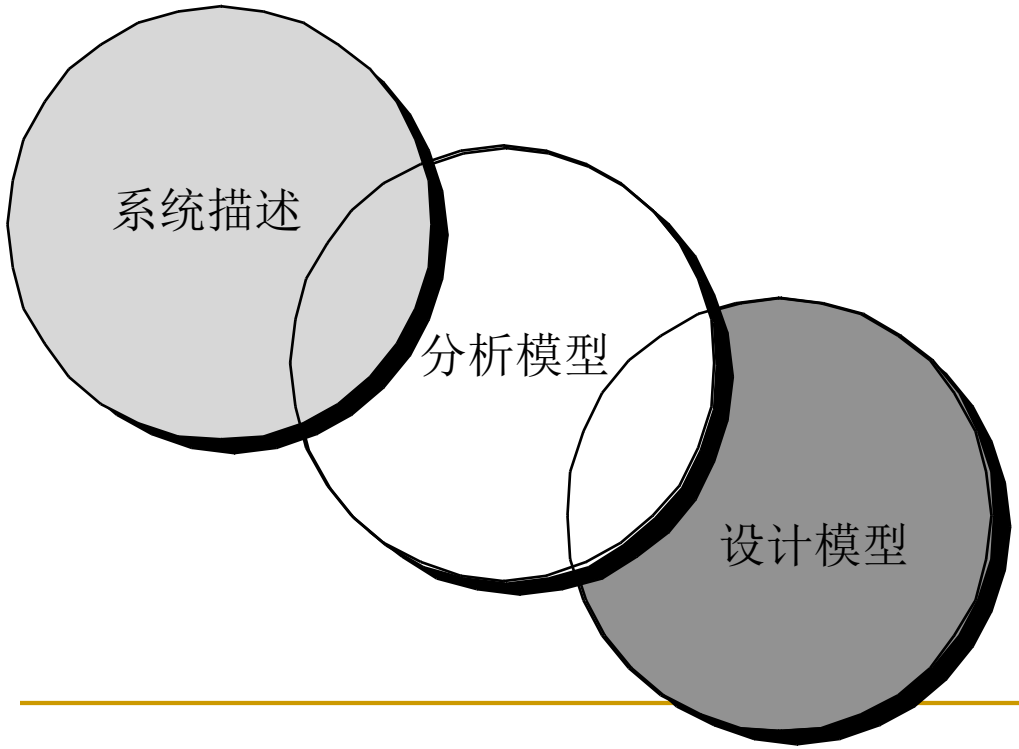
- 验证需求的正确性
  - **一致性（Consistency）**：文档中的需求不应该冲突，包括相互矛盾的约束，及同一系统功能的不同描述
  - **完整性（Completeness）**：SRS应包括用户需要的每一个功能或性能
  - **现实性（Realism）**：现有的软硬件技术可以在系统预算范围内实现需求，应检查系统开发预算和进度
  - **有效性（Validity）**：需求确实满足用户的实际需要
- 需求验证技术
  - 人工审查
  - 原型

outline



总体目标

- 3个目标
  - 描述客户需要什么
  - 为软件设计奠定基础
  - 定义在软件完成后可以被确认的一组需求



需求模型的所有元素都可以直接跟踪到设计模型

需求建模方法

- 每个模型从不同的**视角**描述系统
  - **交互视角**：对系统与**用户**间的交互，以及系统中构件间的交互建模
  - **结构视角**：对系统的**组织结构**建模
  - **行为视角**：对系统的**动态行为**建模