



西安电子科技大学  
XIDIAN UNIVERSITY

# 获取软件需求





# 内容

---

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



# 1.1 何为获取软件需求?

---

- 软件需求不会无缘无故的产生，也不应是无源之水、无本之木，任何软件需求都有其出处以及动机，即它从何而来、目的是什么
- 获取软件需求是指得到或产生软件需求
- 获取软件需求是需求分析的基础，是软件开发的首要工作

# 软件需求从何而来?

---

## □ 软件开发的动机

- ✓ 解决特定行业和领域的问题，体现意义和价值
- ✓ “MiNote” 是为了帮助用户编写和管理便签； “12306” 是为了帮助旅客解决购买火车票费时费力的问题； “ElderCarer” 是为了解决空巢老人在家无人看护的问题。
- ✓ 需要明确软件所关注和欲解决的问题是什么

## □ 软件利益相关方的诉求

- ✓ 软件系统的受益者，软件需求的提出者
- ✓ 利益相关方有哪些，他们会对软件提出什么样的期望和要求
- ✓ “12306” 的利益相关者是旅客，服务于旅客的期望和要求
- ✓ 与软件产生交互的其他系统也是利益相关者，也会对软件提出各种要求



# 软件开发者也可以充当软件的利益相关方

---

□许多软件系统的需求来自软件工程师，而非最终的用户

✓微信、12306等等

□他们充当软件的用户或客户，构思和提出软件需求

✓原因是找不到实际的用户来提出需求

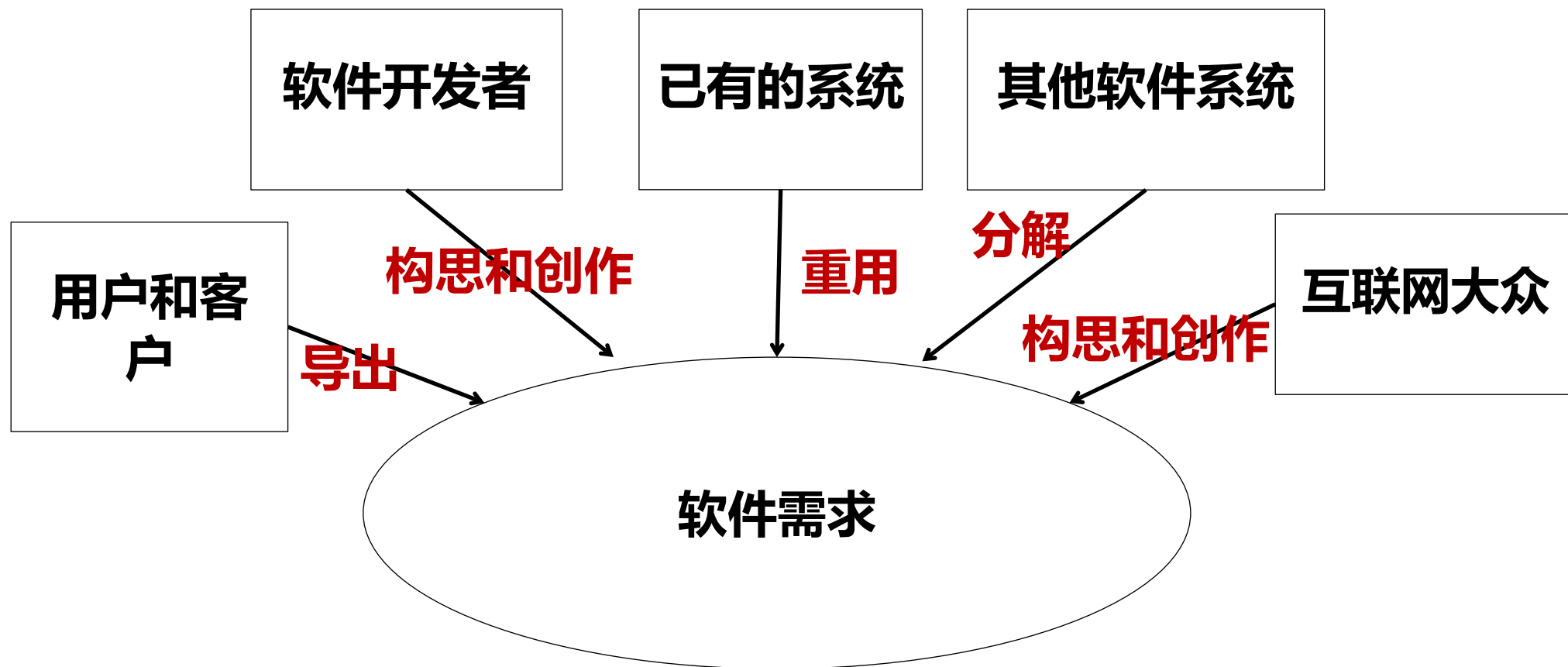
□开源软件的许多需求来自软件开发者，而非最终用户

✓提出软件的功能性构思Issues



## 1.2 获取软件需求的方式

获取软件需求是要解决软件需求从无到有的问题，是一项创新性工作。





# 获取软件需求的困难

---

## □需求想不清、道不明，

- ✓ “用得越多，需求越多”，需求想不清易导致需求不足
- ✓ **Who**，谁会关心该项软件需求，他们有何特点和诉求
- ✓ **What**，软件需求的内涵是什么
- ✓ **Why**，为什么需要软件需求，想解决什么样的问题
- ✓ **Where**，软件需求归属于哪些子系统
- ✓ **How**，软件需求包含哪些行为，它们是如何来解决问题的
- ✓ **When**，什么时候需要该项软件需求

# 1.3 获取软件需求方法

---

- 访谈和会议
- 调查问卷
- 现场观摩
- 分析业务资料
- 软件原型
- 群体化方法



# 成立需求分析的联合工作小组

## □小组成员

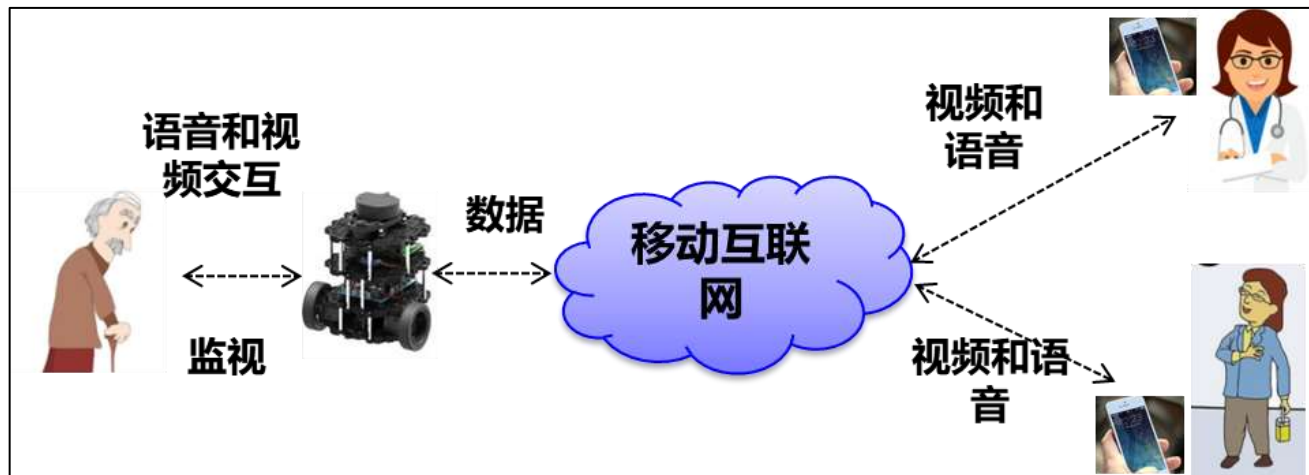
- ✓利益相关者（如用户）
- ✓软件需求分析人员
- ✓质量保证人员等

## □目的

- ✓共同工作
- ✓加强联系
- ✓促进交流
- ✓增进合作



- 召开相关的会议
- 走访相关的人员



**参加的人员包括：用户、客户、需求分析人员、质量保证人员等等  
从整体到局部，从抽象到具体  
有备而来，有计划的开展**

**适用于软件用户或客户明确且能主动配合需求工程师开展工作的软件项目**

## □设计调查问卷

- ✓了解业务流程、实际操作过程
- ✓用户的期望和要求
- ✓要有针对性，每项问题都描述准确，尽可能用选择题

## □对谁进行调查

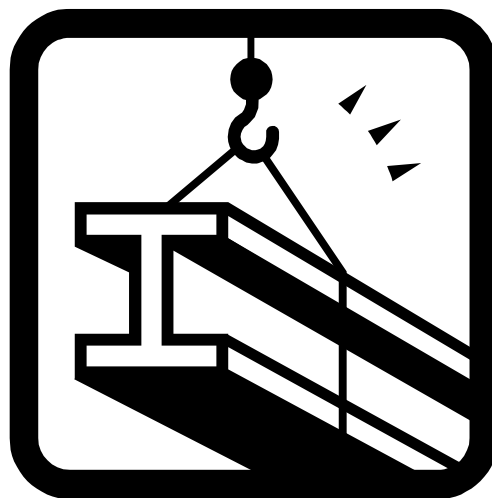
- ✓应用领域中扮演不同角色的用户

## □如何利用调查问卷

- ✓从调查问卷中了解业务流程
- ✓从调查问卷中掌握用户需求



- 业务过程、步骤和输出
- 业务工作流程及细节



# 分析业务资料

- 收集重要的资料
- 阅读和分析与业务相关的文档和资料
- 学习业务流程和领域知识，掌握领域术语，了解业务细节





# 软件原型

---

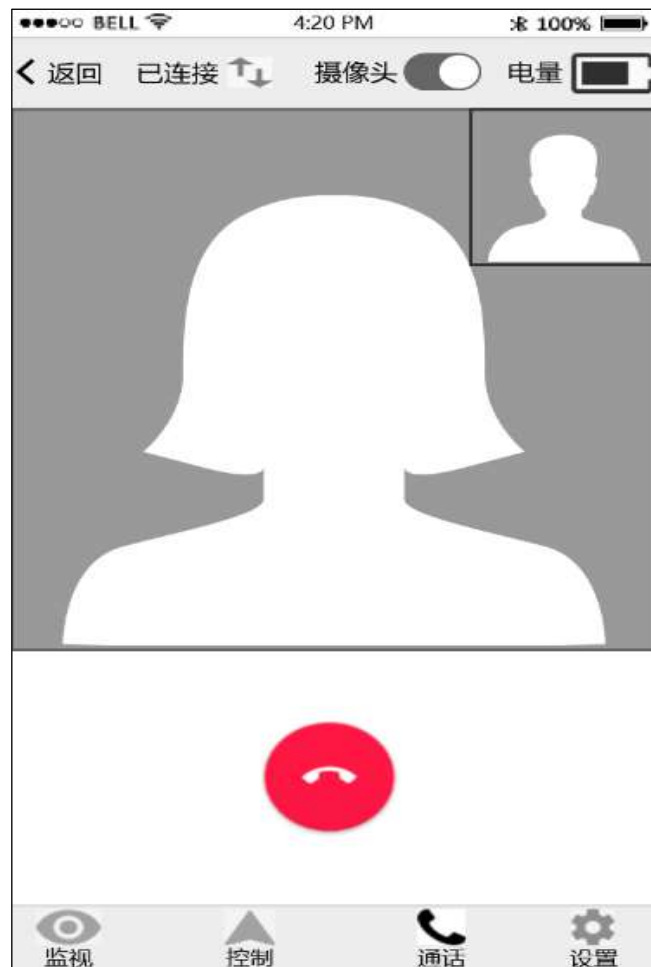
□需求工程师根据用户初步需求描述，快速构造出一个**可运行的软件原型**

- ✓展示业务操作流程以及每一个步骤用户与软件之间的交互
- ✓用户可以通过操作和使用该软件，分析需求工程师是否正确地理解了他们所提出的软件需求，发现软件原型所展示的软件需求中存在的问题，导出尚未发现、新的软件需求

□软件原型作为需求工程师和用户之间的交流媒介，有助于直观地展示软件需求，激发用户投入到需求讨论和导出之中，因而是一项极为有效的需求获取和分析方法

# 示例：软件原型

## □ “空巢老人看护软件” 的原型





# 群体化方法

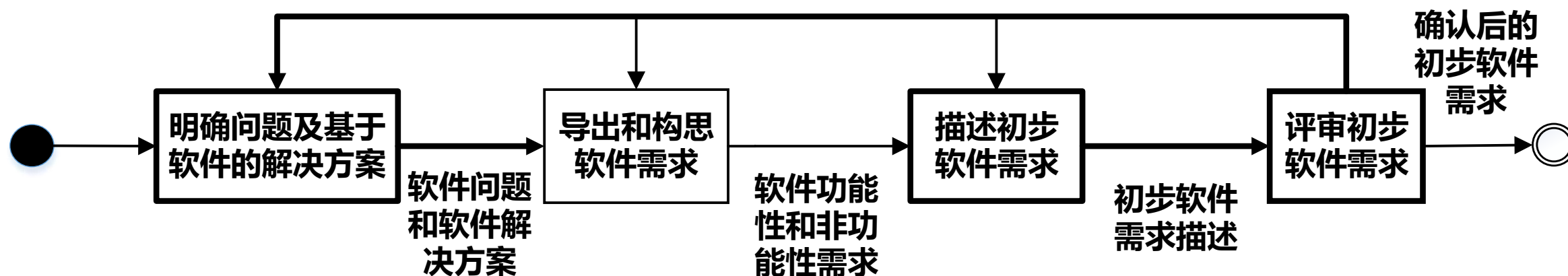
---

- 让互联网上的**海量和开放群体**参与到获取软件需求的工作中来，提出他们对软件需求的想法，并通过组织、汇聚和筛选，从中遴选出有价值的软件需求
- **集思广益**，吸纳更多的人参与到需求创作和构思之中，有助于获得超出开发团队和用户常规想定的软件需求





# 1.4 获取软件需求的过程





# 内容

---

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求





## 2.1 软件的目的

---

- 每一个软件都试图去解决特定领域中的问题，并提供**基于软件的问题解决方案**
- 软件需求必须服从和服务于软件**欲解决的问题**，只有这样软件需求才有意义和价值

# 明确软件要解决的问题

---

## □ 清晰地界定软件欲解决什么样的问题

- ✓ 与特定领域及其业务相关联
- ✓ 或提高业务工作效率，或解决业务瓶颈问题，或提升业务服务水平和质量等等

## □ 示例

- ✓ “12306” 软件与铁路旅客服务这一领域及业务相关联，国家铁路集团有限公司投资研制该软件的目的是要改变落后的旅客服务和业务模式，提高旅客服务质量，降低旅客服务成本



# 定义软件欲解决问题

□开展**调研分析**，切忌拍脑袋凭空想问题

✓铁路旅客存在买票难的问题

□不断**反复论证**，寻找适合软件解决的问题

✓通过软件来支持旅客在线上买票

□寻求**有意义、有价值的**软件问题

✓买票、退票、改签等等





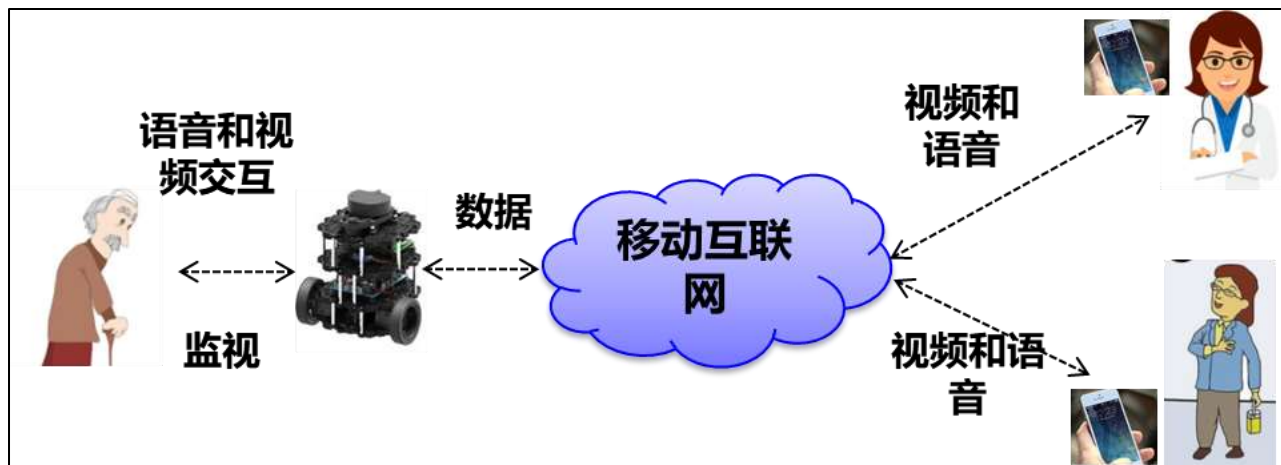
# 明确基于软件的问题解决方案

□在宏观层面寻求基于软件的问题解决方案，不涉及技术细节

□软件可以起到什么作用

✓完成各种复杂计算，作为一种粘合剂来连接不同的设备和系统，实现不同设备和系统之间的交互和协同，从而来解决问题

□发挥软件的优势，为问题解决提供新颖和有效的途径





# 内容

---

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求





## 3.1 识别软件的利益相关方

---

- 软件需求来自于软件的**利益相关方**，要获取软件需求，首先要搞清楚软件系统有哪些利益相关方
- 软件系统的利益相关方可以表现为特定的人群和组织，也可以是一类系统
- 不仅**软件用户或客户**可以是软件的利益相关方，软件的**开发者**也可以成为软件的利益相关方。





# 示例：“空巢老人看护软件”的利益相关方

---

## □老人 (Elder)

- ✓与系统通过语音方式进行交互，命令系统为其完成某些事务

## □医生 (Doctor)

- ✓突发或紧急情景时接受呼叫，与系统进行交互以获取老人状况

## □家属 (Family Member)

- ✓突发或紧急情景时接受呼叫，与系统进行交互以获取老人的状况

## □管理员 (Administrator)

- ✓对软件系统进行必要的配置和管理

## □机器人 (Robot)

- ✓其运动受软件系统的控制



## 3.2 导出软件的功能性需求

---

- 需求工程师可以通过与利益相关方的交互，听取他们对软件的**期望和要求**，从他们那里导出软件需求
- 采用多种方法来导出软件需求
  - ✓包括与用户或客户的面谈、分析业务资料、观察业务流程、进行问卷调查、软件原型等。



# 示例：构思“空巢老人看护软件”功能性需求

---

## □老人的视角

- ✓自主跟随老人，以对老人进行持续跟踪和感知，获取老人信息。
- ✓提醒服务，提醒老人按时服药、身体检查等事宜

## □家属和医生

- ✓监视老人状况，通过智能手机在远端监视老人在家的状况，获得老人的视频、图像和语音等方面的信息
- ✓通知异常情况，将老人的突发异常信息发送给老人家属和医生。
- ✓远程控制机器人，通过智能手机在远端来控制机器人移动，以从不同的角度和距离来获取老人的图像、视频和语音信息
- ✓视频/语音双向交互，实现老人、医生和家属间的视频和语音交互



## 3.3 导出和构思软件的非功能性需求

---

### □非功能性需求包括**软件质量要求**和**软件开发的约束性要求**

- ✓ **质量要求**，如软件运行性能、可靠性、易用性、安全性、私密性等属于外部质量要求，软件可扩展性、可维护性、可互操作性、可移植性等内部要求
- ✓ **约束性要求**，包括开发进度要求、成本要求、技术选型等

### □软件的**非功能性需求**变得越来越重要

- ✓ 在某些情况下它们直接决定了软件是否能用和可用、是否好用和易用、是否高效和可靠运行、是否便于维护和演化等。



# 示例：“空巢老人看护软件”的非功能需求

类别	非功能性需求项	需求描述
性能	EC-Req-Performance-01	所有界面操作的响应时间应小于1s
可靠性	EC-Req-Reliability-01	软件系统每周7天、每天24小时可用；在机器人和网络无故障的前提下，系统正常运行时间的比例在95%以上
	EC-Req-Reliability-02	系统任何故障都不应导致用户已提交数据的丢失。发生故障后，系统需在10分钟内恢复正常使用
易用性	EC-Req-EasyUse-01	老人只需通过语音方式与系统进行交互；家属、医生和管理人员通过操作手机App来使用本系统
	EC-Req-EasyUse-02	用户无需专门培训只需通读安装手册即可完成安装
安全性	认证需求EC-Req-Safety-Authentication-01	所有用户（包括家属和医生）均需通过用户名、密码相结合的方式经系统验证通过后方可使用本软件系统
	权限控制需求EC-Req-Safety-Authorization-001	医护人员只能在老人处于紧急状况、需要求助时可控制机器人查看老人的状况，家属可以在任何时候查看老人的所有信息
运行环境约束	客户端EC-Req-Env-Client-001	客户端APP运行在Android 4.4及以上版本
	服务器端EC-Req-Env-Server-001	服务器端的软件运行在Ubuntu 14.04及以上版本；本软件运行时占用的内存空间不得超过128MB



# 合法的软件需求

---

□并非软件利益相关方提出的每项期望和要求都是软件需求

✓如果他们提出的要求与软件及其欲解决的问题无关、没有实际的意义和价值、或者不存在技术等方面的可行性，那么这些要求不应成为软件需求

□软件需求工程师要有一双“火眼睛”，能够**鉴别和发现真正有意义和有价值的重要软件需求**

# 课程实践创意示例1：无人值守图书馆

## □问题

- ✓找寻图书困难

## □创意

- ✓线上借阅
- ✓线下机器人获取
- ✓自动找寻



线上基于智能终端的借阅



线下基于机器人的服务



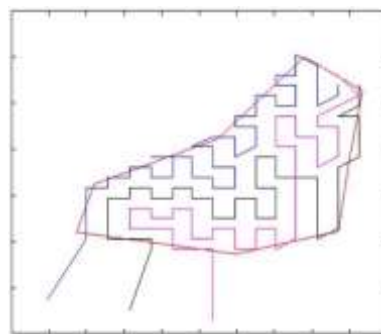


## □问题

- ✓快速定位被埋人员

## □创意

- ✓借助手机定位
- ✓采用无人机编队搜索
- ✓自动生成位置







# 内容

---

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



## 4.1 描述初步软件需求

---

### □初步的软件需求

- ✓之所以称之为是**初步的**，是因为这些软件需求还很粗略，只是一个初步的需求轮廓，不够具体和详尽，可能有遗漏，会存在不一致和相冲突等问题，后续还需要对其开展进一步的精化和分析

### □描述需求

- ✓记录下来、描述清楚，形成相关的软件文档，以便于不同人员（如需求工程师、用户、客户等）之间的交流和讨论，及时发现需求理解上存在的偏差，支持后续的需求分析工作



# (1) 自然语言描述

---

## □ 自然语言是最为常用的需求描述手段

- ✓ 描述软件的**功能性需求、质量需求和开发约束需求**等
- ✓ 详细刻画需求的具体内容和细节
- ✓ 可为各方所理解，便于交流和讨论



# 示例：用自然语言描述的软件需求

---

## □功能性需求描述

- ✓软件系统需要对老人在家的状况进行分析，以判断是否出现突发异常情况。一旦出现异常情况，就需要通知老人家属和医生

## □质量需求描述

- ✓老人通过语音方式与系统进行交互，系统正确理解老人语音指令的比率应达到90%以上

## □软件开发约束需求描述

- ✓客户端App软件须运行在Android 4.4及以上版本的操作系统



# 自然语言描述的不足

---

## □不具体

- ✓如老人的信息是指什么，上述自然语言描述没有刻画清楚

## □不准确

- ✓何为突发异常情况，上述自然语言描述没有定义明确

## □有二义

- ✓不同的人看完描述后可能会有不同的理解

## □不直观

- ✓很难从中厘清软件系统到底有哪些功能性需求和非功能性需求，这些需求之间存在什么样的关系。



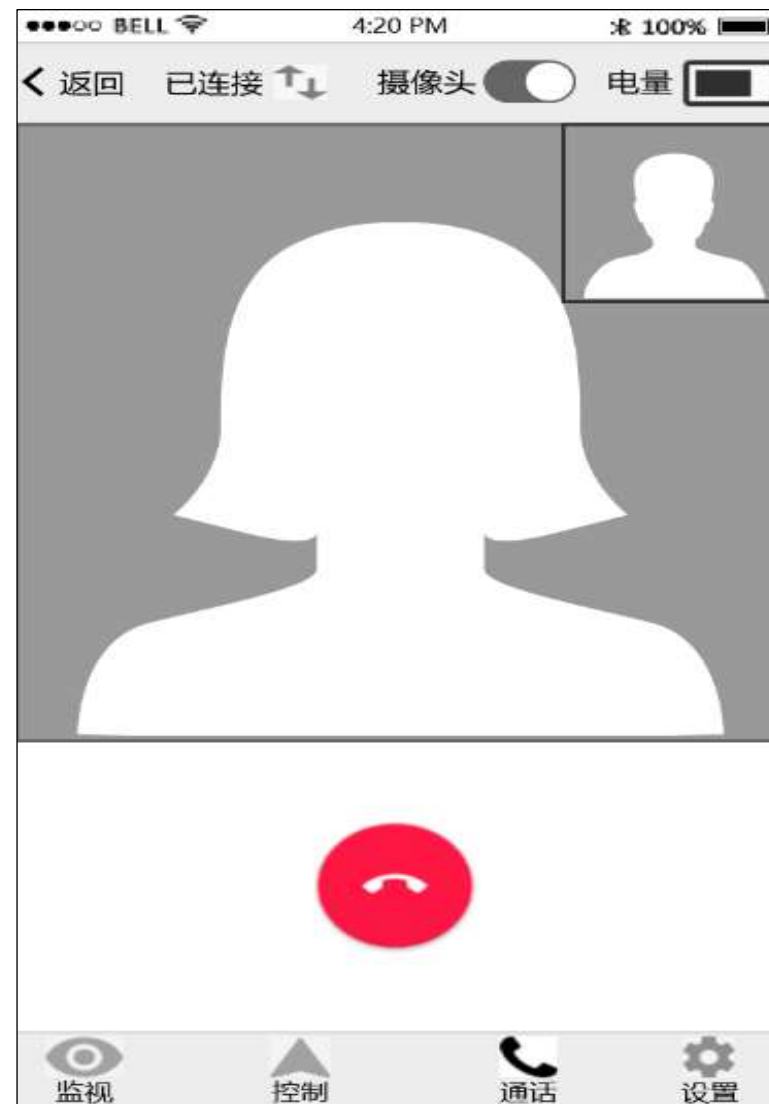
## (2) 软件原型描述

### □优势

- ✓ **直观、可展示和可操作**
- ✓ 有助于需求工程师与用户或客户之间的交流和沟通，便于在操作和使用软件原型的过程中帮助用户和客户确认和导出软件需求

### □不足在于

- ✓ 主要以操作界面的形式展示软件需求的梗概，主要是软件与用户之间的输入和输出，业务的大致流程，**无法描述软件需求的具体细节**



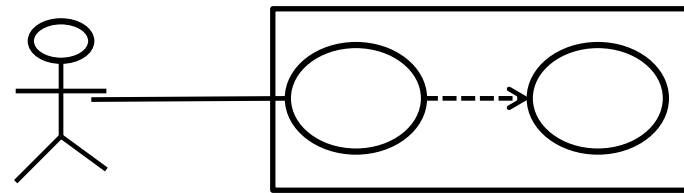
## (3) 用例图描述

### □用途

- ✓ 描述软件系统的**边界**以及软件**外部使用者**所观察到的**系统功能**
- ✓ “观察到”是指外部使用者与系统存在**交互**，即信息输入和输出

### □图的构成

- ✓ **执行者** (Actor) : 系统的外部使用者
- ✓ **用例** (Use Case) : 观察到的功能和交互
- ✓ **边** : 执行者与用例间、用例之间、执行者之间的关系



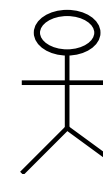
# 执行者(Actor)

## □何为执行者

- ✓ 系统**之外的实体**，他们使用软件系统功能、与软件系统交换信息
- ✓ 可以是一类**用户**，也可以是其他**软件系统或物理设备**

## □执行者是UML中的类

- ✓ 代表一类用户或者外部实体，而非具体的对象实例

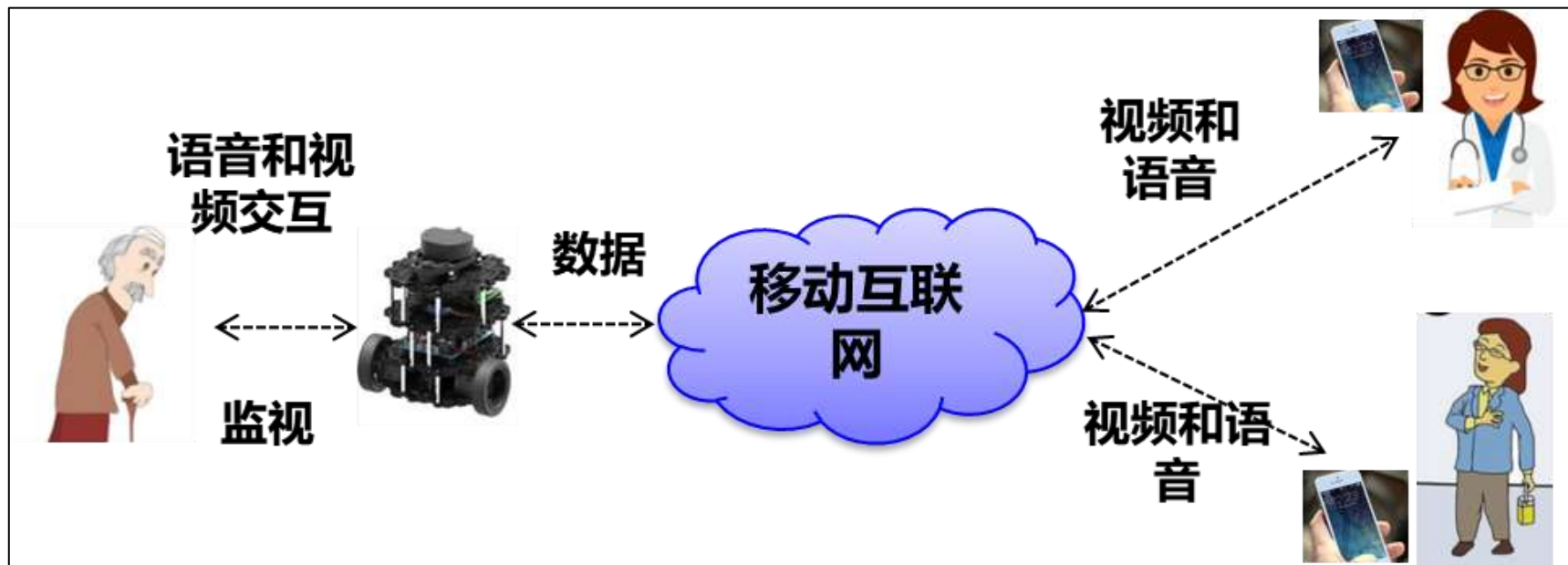


**Actor**  
**Name**



# 示例：“空巢老人看护软件”的执行者

- 老人
- 医生
- 家属
- 机器人
- 管理员
- ...



**执行者通常对应于软件系统的利益相关方**

# 用例(Use Case)

## □用例

- ✓表示执行者为达成一项相对独立、完整的业务目标而要求软件系统完成的**功能**



Use Case

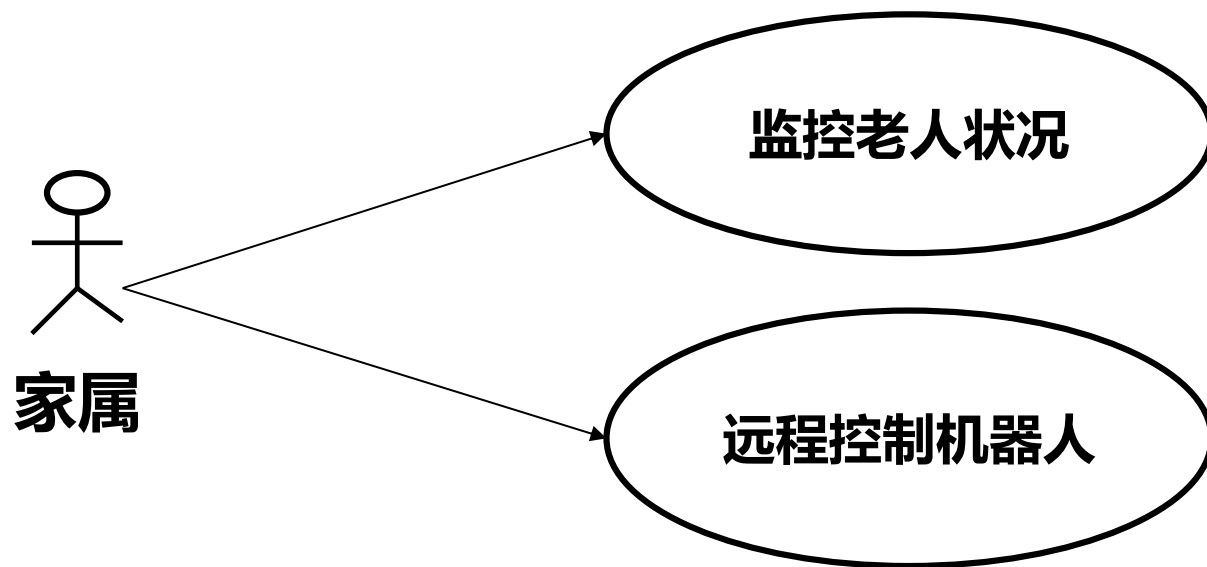
## □用例的表现形式

- ✓表现为执行者与系统之间的**业务交互动作的序列**
- ✓对于执行者而言，交互目的或者效果在于达成其业务目标
- ✓对于待开发系统而言，交互的过程即是某项相对独立、完整的外部可见功能的实现过程

# 示例：用例

## □ “家属” 执行者所看到的用例

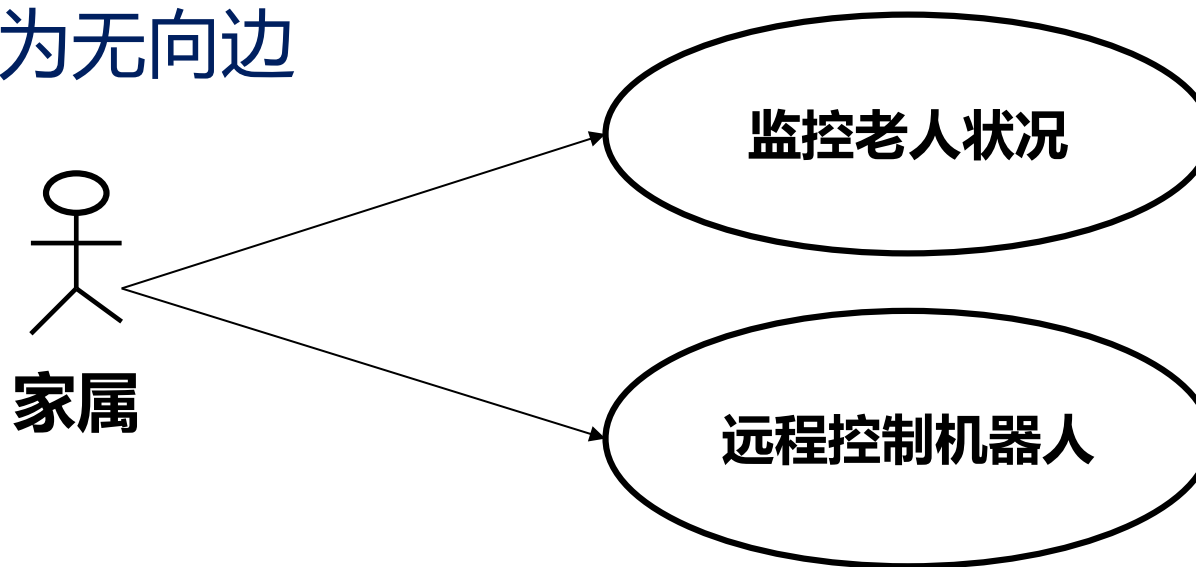
- ✓ 监控老人状况
- ✓ 远程控制机器人



# 执行者与用例间的关系

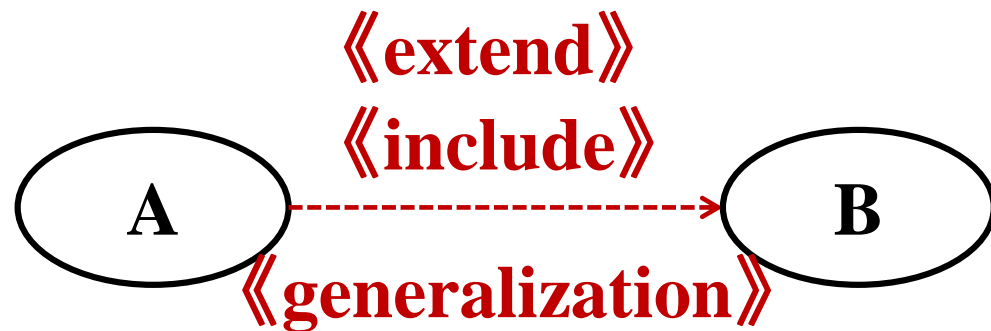
## □执行者与用例间关系的内涵

- ✓ 执行者与用例之间存在**交互**
- ✓ 执行者触发用例执行，向用例**提供信息或从用例获取信息**
- ✓ 触发用例执行的执行者称为**主动执行者**，仅从用例获取信息的执行者称为**被动执行者**
- ✓ 执行者与用例间的边通常为无向边



## 4.2 用例间的关系

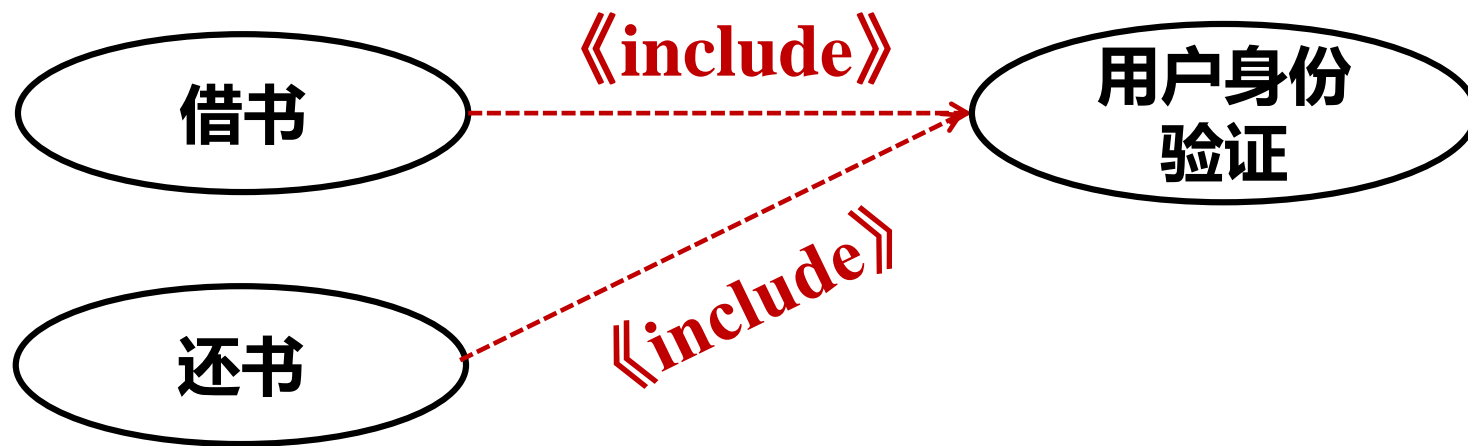
- ① 包含(Include)
- ② 扩展(Extend)
- ③ 泛化(generalization)



# (1) 用例间的关系-包含关系

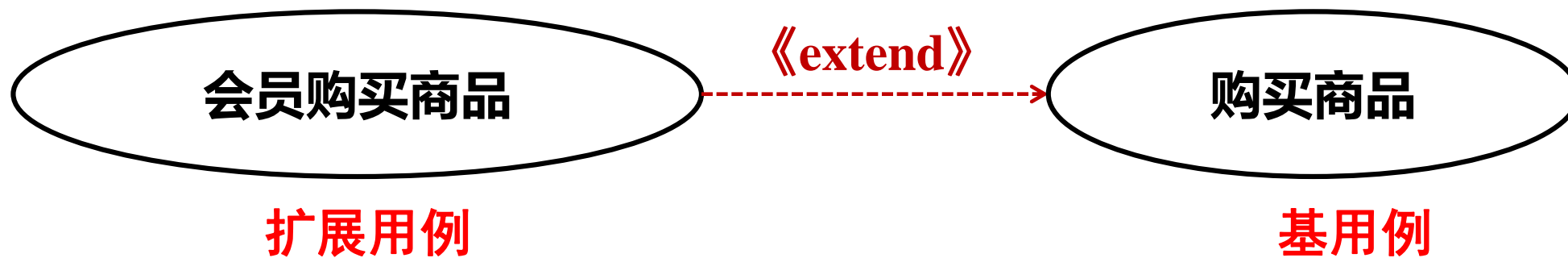
□如果用例B是用例A的某项**子功能**，则称用例A包含用例B

- ✓包含是一种 “uses a”关系，表示一个用例在执行过程中需要用到另一个用例的功能，被包含的用例功能是包含用例功能的一部分。
- ✓体现了功能分解和组织的思想



## (2) 用例间的关系-扩展关系

- 扩展是一种 “extends a” 关系，表示一个用例（扩展用例）在特定条件下对另一个用例（基用例）进行扩展，增加新的功能或行为，但不改变基用例原有的功能
- 用来区隔正常业务处理功能和带有例外处理的功能
  - ✓ 避免例外处理逻辑搅乱或湮灭正常处理逻辑
  - ✓ 体现了功能增强



# 泛化和扩展的区别

对比维度	泛化 (Generalization)	扩展 (Extend)
定义	父子用例间的 <b>继承关系</b> ，子用例是父用例的特化版本。	基础用例在特定条件下 <b>可选地触发扩展用例</b> 。
触发条件	无条件继承，子用例直接替代父用例执行。	有条件触发（如“书籍不可用”时）。
方向性	单向的层级关系（父→子）。	基础用例指向扩展用例（基础用例是主体）。
用例独立性	子用例必须实现父用例的所有步骤，不可独立存在。	扩展用例是独立的，仅在条件满足时附加到基础用例。
设计目的	复用共性流程，处理 <b>不同角色的差异化需求</b> 。	处理基础用例中的 <b>可选分支或异常场景</b> 。
示例核心逻辑	学生借书 vs 普通借书（规则不同，但流程相似）。	借书时书籍不可用 → 触发预约。





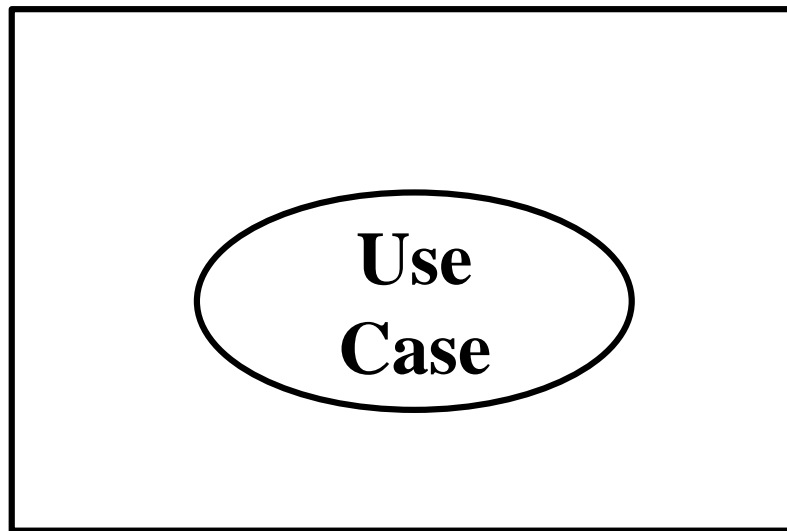
## (4) 边界框

### □表示整个软件系统或子系统的边界

- ✓边界框内的用例构成了系统或子系统的内容，如用例
- ✓外面的是系统之外的执行者

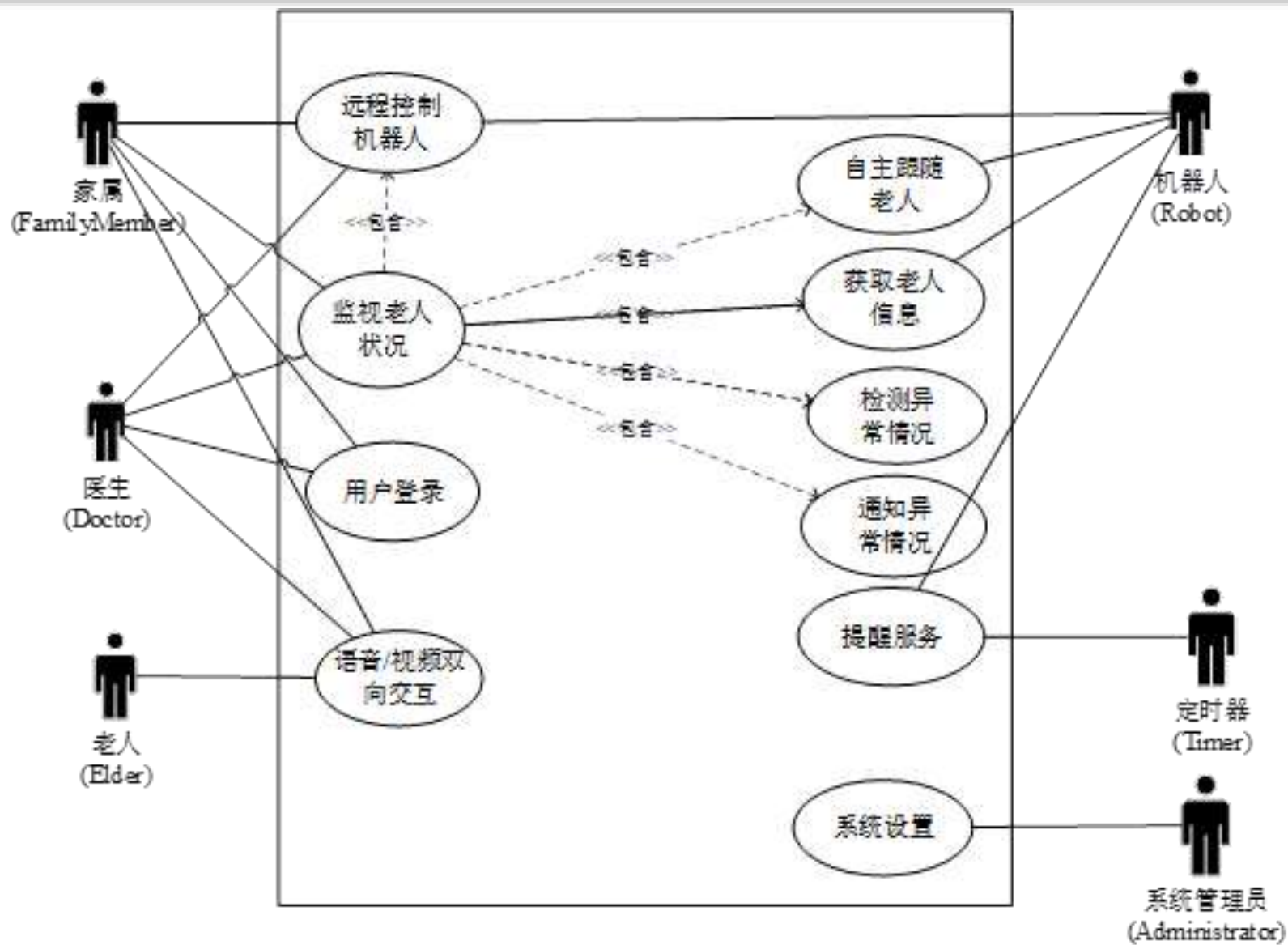


**Actor  
Name**





# 示例：“空巢老人看护软件”的用例图





# 示例：用例的进一步详细描述

---

- **用例名：** 用户登录
- **用例标识：** UC-UserLogin
- **主要执行者：** 家属、医生
- **目标：** 通过合法身份登录系统以获得操作权限
- **范围：** 空巢老人看护软件
- **前置条件：** 使用App软件之时
- **交互动作：**
  - ✓ 用户输入账号和密码
  - ✓ 系统验证用户账号和密码的正确性和合法性
  - ✓ 验证正确和合法则意味着登录成功



## 4.3 撰写“初步软件需求描述”文档

---

- 软件背景介绍
- 欲解决的问题
- 软件解决方案
- 软件的功能性需求描述
- 软件的非功能性需求描述
- 可行性及潜在风险



# 内容

---

## 1. 获取软件需求的方法

✓ 软件需求的方式和过程

## 2. 明确问题及软件解决方案

✓ Problem → Software Solution

## 3. 导出和构思软件需求

✓ 如何从利益相关方导出和构思软件需求

## 4. 描述初步的软件需求

✓ 自然语言描述和可视化建模

## 5. 确认和验证初步软件需求



## 5.1 输出的软件制品

---

### □软件原型

- ✓以**可运行软件的形式**展示了软件的业务工作流程、操作界面、用户的输入和输出等方面的功能性需求信息

### □软件用例模型

- ✓以**可视化图形符号的方式**刻画了软件系统的执行者、边界、用例以及它们之间的相互关系，描述了软件的功能性需求

### □软件需求文档

- ✓以**自然语言的方式**描述了初步软件需求，包括功能性和非功能性的软件需求

## 5.2 评审初步软件需求

---

- **中肯性**，软件问题是否反映实际问题，是否有意义和价值
- **合理性**，基于软件的解决方案是否科学和合理。
- **完整性**，软件需求是否覆盖了利益相关方的期望和要求
- **必要性**，每一项软件需求是否有必要
- **溯源性**，每一项软件需求是否都有其来源
- **准确性**，描述是否清晰和准确地反映了软件需求的内涵
- **正确性**，是否正确反映该需求提出者真实想法和关注点
- **一致性**，软件需求文档、用例模型以及软件原型对软件系统需求的表述（包括术语等）是否一致



## 5.3 软件需求可行性分析 (1/2)

---

### □技术可行性

- ✓需要哪些技术，相关的技术是否成熟，现有技术能否支撑软件需求的实现，软件项目团队是否已经掌握了某些关键技术等等

### □设备可行性

- ✓基于软件的解决方案需要哪些设备或系统，软件项目团队是否已经具备这些设备和系统

### □进度可行性

- ✓软件用户或客户对软件提出什么样的进度要求，针对开发团队的人力资源及技术水平，能否遵循进度约束开发出满足这些需求的软件产品





# 软件需求可行性分析

---

## □成本可行性

- ✓ 基于软件项目成本约束，软件项目团队能够开发出满足软件需求的产品出来

## □商业可行性

- ✓ 软件需求是否有商业价值，能否获得用户的青睐以及市场的认可，针对这些软件需求的投入能否获得预期的收益

## □社会可行性

- ✓ 当前软件系统已经成为国家和社会的重要组成成分，需要从社会的角度，评估软件的各项需求是否违背社会道德、文化伦理、法律法规、行业标准等，或者与它们是否存在冲突

## □ 软件需求来自于软件利益相关方

- ✓ 导出和构思软件需求

## □ 循序渐进地构思和导出软件需求

- ✓ 问题、基于软件的解决方法、软件利益相关方

## □ 描述初步的软件需求

- ✓ 自然语言、UML建模（用例图）、软件原型

## □ 验证和确认

- ✓ 多方从多个方面来评审初步的软件需求



# 综合实践

---

## □任务：构思待开发软件系统的需求

### □方法

- ✓构思软件需求，从分析软件的利益相关方入手，站在他们的视角来构思软件需求，以解决软件问题；要从规模、创新、可行性等多个方面分析所构思的软件需求，确保其质量，满足实践的基本要求；借助于UML用例图来刻画初步软件需求

### □要求

- ✓要求所构思的需求要有意义和价值、软件功能有新意、各项软件需求存在技术可行性，确保整个软件系统有一定的规模，讨论每一项软件需求的可行性

## □结果：初步软件需求的UML模型和软件文档