

获取软件需求

内容

1. 获取软件需求概述

- ✓ 获取需求的方法
- ✓ 获取需求的过程

2. 明确问题及解决方案

- ✓ Problem → Software Solution

3. 导出和构思软件需求

4. 描述初步的软件需求

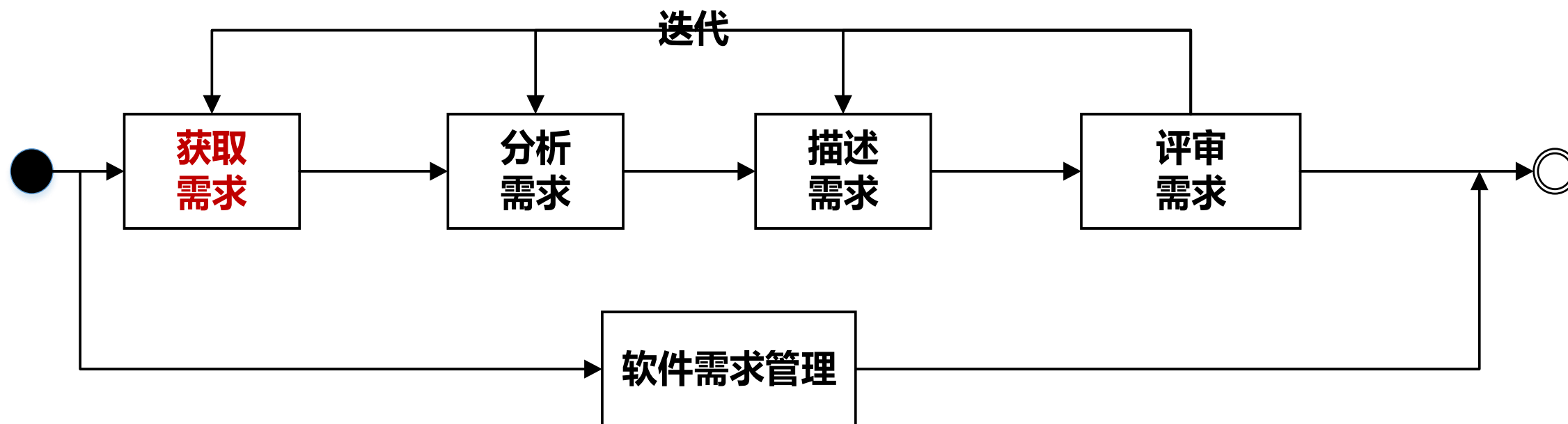
- ✓ 自然语言描述和可视化建模

5. 确认和验证初步软件需求



1.1 概述

- ❑ 获取软件需求是需求分析的**基础**
- ❑ 需求获取可能是软件开发中最困难、最关键、最易出错的任务。



软件需求获取的困难

□需求想不清

- ✓项目初期，用户往往提不出太多需求或需求较模糊，直至看到部分产品时，才有更多的想法。
- ✓软件系统是全新的，没有可模仿和参照的对象。

□缺乏专业知识

- ✓需求人员缺乏足够的行业知识，难以准确理解客户的需求

□需求道不明

- ✓不清楚软件需求的内涵，难以清晰、准确地讲表达软件需求



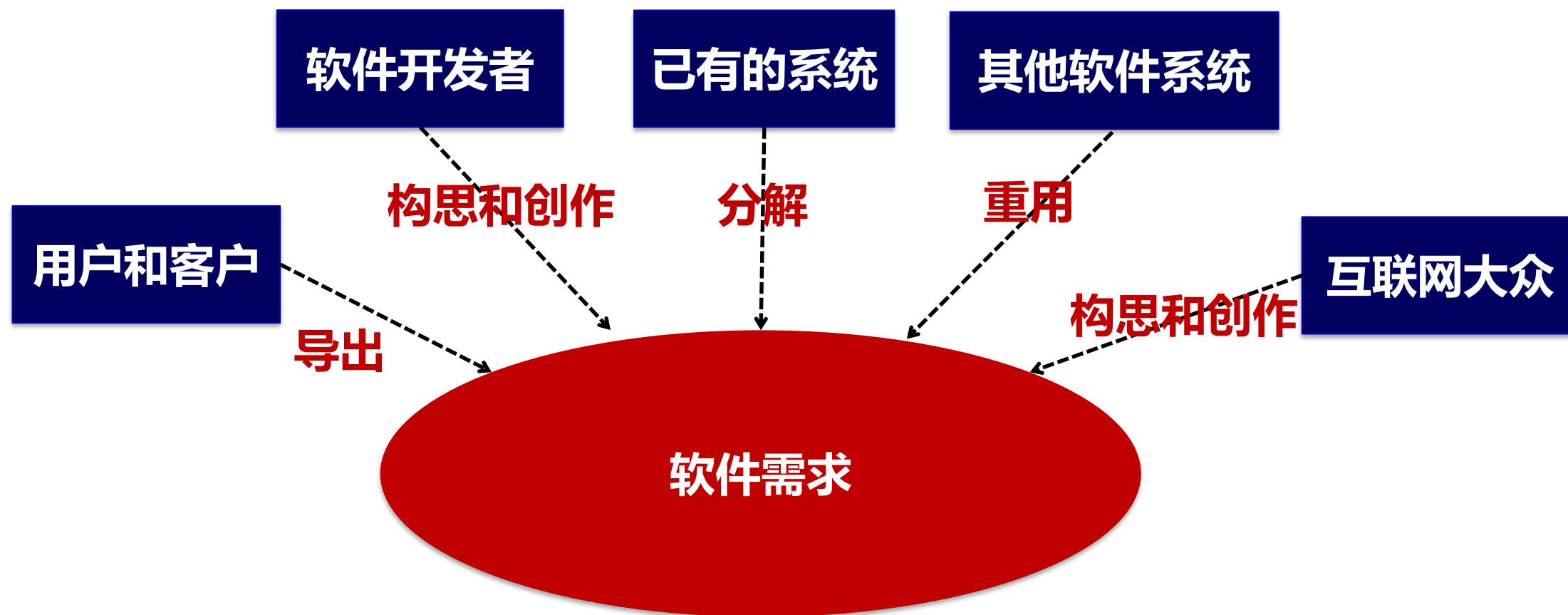
完整软件需求的构成

- Who，谁会关心该项需求
- What，需求的内容是什么
- Why，为什么会需要该软件需求
- Where，When，产生该需求的时间、地点、环境等
- How，如何确认该需求被满足了（即验收标准）

需求编号： 2017. 10. 15	需求类型： 功能性需求-导航
来源（Who） 客服用户反馈	
场景（Where，When） 使用小黄车骑行的路上	
描述（What） 用户在骑行过程中，可以在小黄车 app 界面搜索目的地，规划最佳路线，对到达指定目的地进行导航。	
原因（Why） 很多用户在打开小黄车 app 扫码后，不知道骑行目的地的路线，需要打开第三方导航软件（百度地图、高德地图等）进行导航，十分不便，	
验收标准（How） 用户能根据导航路线正确到达目的地	需求重要性权重（How much）

要说清楚一项软件需求，就需要说清楚有关需求的以上6个方面内容

1.2 软件需求的来源



软件需求来自不同的**利益相关方**

软件需求的来源

- 不同的软件，需求来源可能不同
- 大多数需求来自于**用户或客户**
- 也有许多软件需求来自**软件工程师**，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
 - ✓ 示例：如开源软件，微信软件，12306软件等等
- **开源软件**的许多需求则来自**软件开发**者
 - ✓ 提出软件的功能性构思及需求，并以Issues的形式提出

1.3 获取软件需求方法

- 访谈和会议(Interview)
- 调查问卷(Questionnaire)
- 现场观摩
- 分析业务资料
- 软件原型
- 群体化方法
- 大脑风暴构思 (Brainstorming)
-

访谈和会议

- 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

- 拟定会议议程

- ✓ 确定会议计划
- ✓ 准备调研提纲

- 指定“调解人”控制会议

- ✓ 可以是用户或需求人员
- ✓ 需求人员引导和把握调研的重点和方向
- ✓ 从业务流程入手，重点关注其中的功能需求（核心是IPO）

从访谈和参与会议的对象中获得软件需求

调查问卷

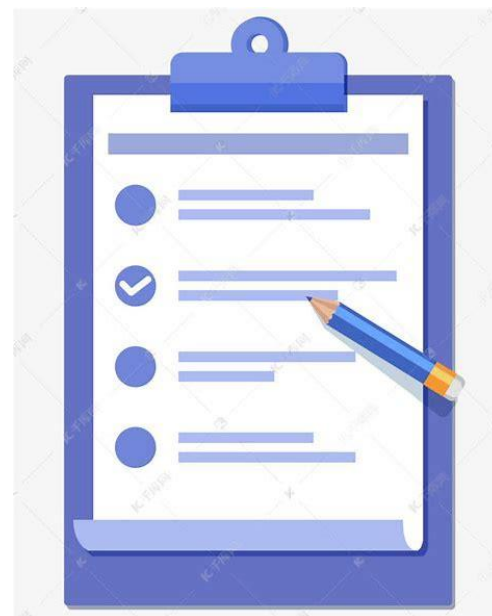
□设计调查问卷

- ✓业务流程、操作过程
- ✓用户的期望和要求

如设计一个问卷，调查车站旅客对12306软件的期望和要求

□对谁进行调查

- ✓应用领域中**不同角色**的用户

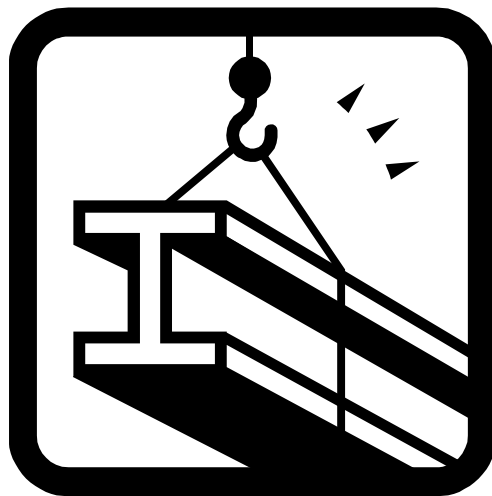


从调查问卷的回答中获得软件需求

现场观摩

- 业务的过程、步骤和输出
- 业务的工作流程及细节

如观摩售票窗口的购票过程




从现场观摩中了解业务流程，进而获得软件需求

分析业务资料

- 收集重要的业务资料
- 阅读和分析业务资料

如银行国际交易的业务材料

火车调度的业务资料



境外汇款申请书
APPLICATION FOR FUNDS TRANSFERS (OVERSEAS)

致： 中国银行
TO: BANK OF CHINA

日期
Date

☐ 电汇 T/T ☐ 票汇 D/D ☐ 信汇 M/T

发电等级
Priority ☐ 普通 Normal ☐ 加急 Urgent

申报号码 BOP Reporting No.		□□□□□□ □□□□ □□ □□□□□□ □□□□			
20	银行业务编号 Bank Transac. Ref. No.	T/T	收电行/付款行 Receiver / Drawn on	BANK OF CHINA	
32A	汇款币种及金额 Currency & Interbank Settlement Amount		金额大写 Amount in Words		
其中	现汇金额 Amount in FX	账号 Account No./Credit Card No.			
	购汇金额 Amount of Purchase	账号 Account No./Credit Card No.			
	其他金额 Amount of Others	账号 Account No./Credit Card No.			
50a	汇款人名称及地址 Remitter's Name & Address				
<input type="checkbox"/> 对公 组织机构代码Unit Code□□□□□□□□-□		<input type="checkbox"/> 对私	身份证件号码 Individual ID NO.		
			<input type="checkbox"/> 中国居民个人Resident Individual <input type="checkbox"/> 中国非居民个人Non-Resident Individual		



通过阅读业务资料来了解业务及其流程，进而获得软件需求

头脑风暴

- 将团队成员组织在一起，非正式、开放地、甚至没有明确主题的**散漫讨论**，从中**捕捉软件需求**的灵感和认识
- 该方法适合于那些需要对软件需求进行**开放构思和自由创作**的软件系统

- **问题定义**：明确讨论的问题。
- **自由讨论**：团队成员自由讨论和交流，提出各种创意和想法。
- **创意整理**：整理团队成员提出的创意，找出可行的方案。

- 胡玮炜与她的摩拜单车



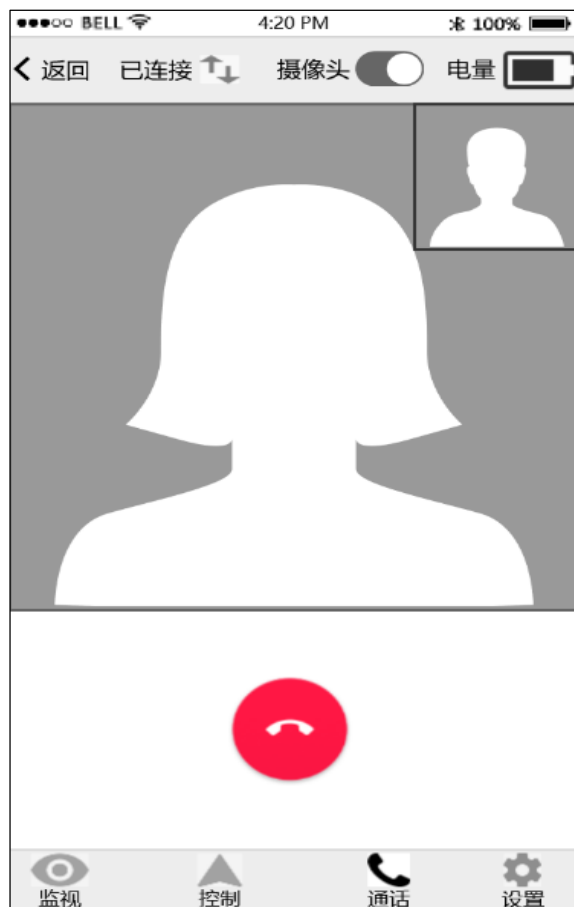
软件原型

- 没看到实际系统前，用户自己都不知道需要什么样的系统
- 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
 - ✓ 展示业务**操作流程** 以及 用户与软件之间的**交互**
- 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**
 - ✓ 用户通过使用原型，可以发现软件需求中的问题，导出尚未发现、新的软件需求

通过软件原型有助于帮助用户导出他们所需的软件要求

示例：基于软件原型来获取软件需求

□ “空巢老人看护软件” 的原型



- 让用户使用该软件原型
- 听取用户对原型的意见和建议
- 用户意见就是潜在的软件需求

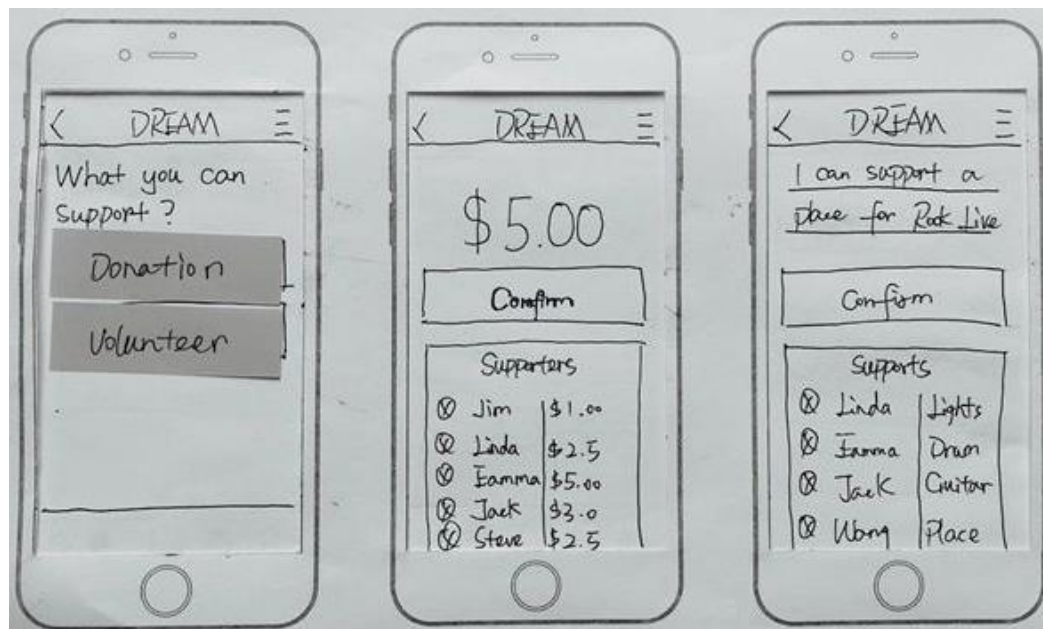
示例：基于软件原型来获取软件需求

□高保真原型

✓ 查看购物车功能原型

添加购物车功能原型

□低保真原型



□原型工具：Axure，墨刀，摹客

群体化方法

- 让互联网上的**海量和开放群体**参与获取软件需求的工作，提出他们的想法，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
- 开源软件的需求就是采用群体化的方法来获取的

依托社会化群体来集体构思软件的需求，集思广益

场景分析法

□场境分析法是工程师通过**模拟用户的使用场景**，分析问题和需求的方法

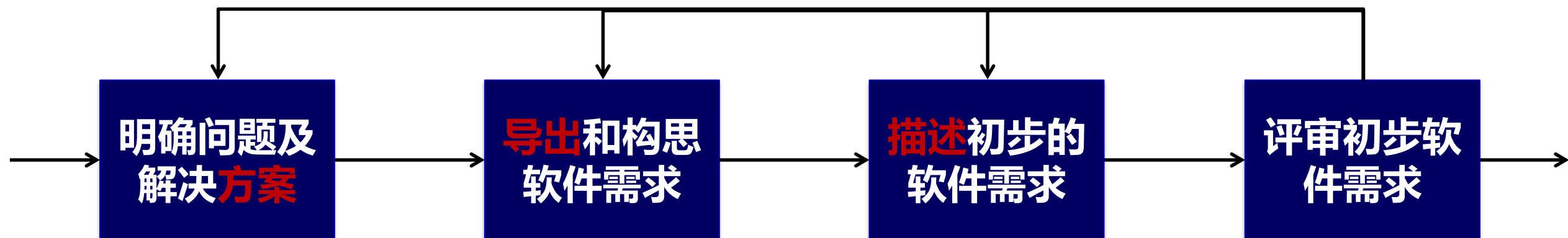
□场景分析的核心是：**角色、场景、方案**



需求场景分析表		
角色	场景（时间、地点、动机）	方案（途径+流程）
A类用户	情况1，要做事情1（而产生了该需求）	方案1：方式&流程1 方案2：方式&流程2
	情况2，要做事情2（而产生了该需求）	方案1：方式&流程1 方案2：方式&流程2

角色	场景	方案	用户占比
订票用户	购买火车票（目的地车票充足）	正常订票-支付	70%
	购买火车票（没有直达车票，但有中转到达方案）	推荐中转方案-按中转方案生成多订单-多订单支付	20%
	购买火车票（没有任何车票）	推荐旅店-生成第二天订单-支付	10%

1.4 获取软件需求的过程



内容

1. 获取软件需求概述

- ✓ 获取需求的方法
- ✓ 获取需求的过程

2. 明确问题及解决方案

- ✓ 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 定义问题



- 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**
- 软件需求必须服务于软件**欲解决的问题**，这样的需求才有意义和价值

定义问题

□开展调研分析,发现(主要)问题

✓铁路旅客存在买票难、候车环境差等问题

□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、有价值的问题

✓买票、退票、改签、打印等

✓儿童手表关注的痛点：安全问题、通话问题、防沉迷问题等

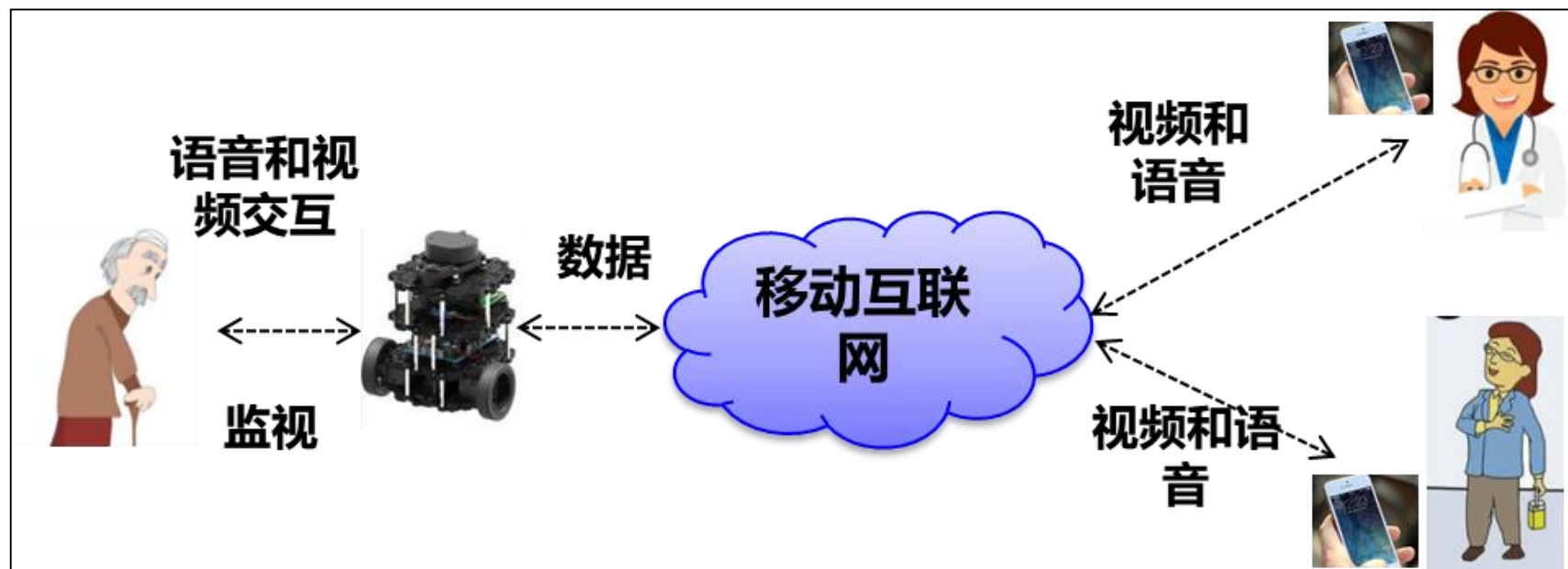


2.2 提出多种解决方案

- **解决方案**就是**针对特定问题的解决思路**，不同方案有不同的构成要素，如设备、软件、框架、通讯等。
- 软件在解决方案中主要起到“**粘合剂**”的作用，通过软件和计算，连接不同的设备和系统，从而解决目标问题。
- **例如：针对员工考勤问题，可选方案包括：**
 - ✓ 硬件加软件方案：指纹考勤机负责采集指纹数据，软件负责统计、分析，计算出勤数据
 - ✓ 全软件方案：地点打卡、WIFI打卡，.....

解决方案分析与选择

□从技术、经济、可靠性、新颖性、先进性等多方面对方案进行评价和选择。

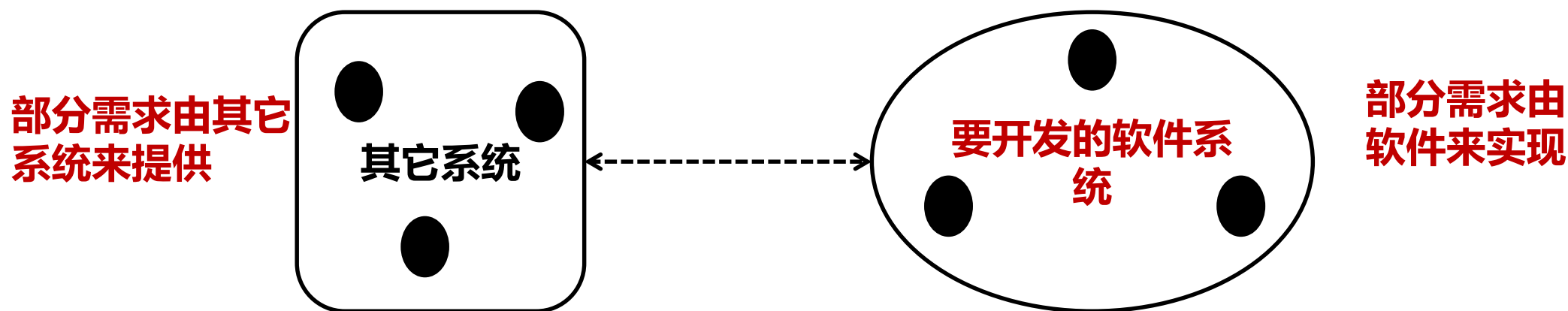


上述方案中，不依靠保姆进行看护，而是借助于软件来控制机器人来进行看护

明确软件的范围和边界

□描述软件的**范围**，确定软件的**边界**

- ✓软件的**范围**说明了**软件需要完成的功能**
- ✓软件的**边界**描述了**软件的界限**，即哪些需求由本软件完成，哪些需求由其他设备和系统来完成



思考和讨论

- 高速不停车收费ETC系统、车库不停车收费系统有哪些可选的解决方案？
- 每种方案的优缺点是什么？
- 每种方案中软件的边界和范围是什么？



只有先确定方案，才能为下一步导出需求打下基础

内容

1. 获取软件需求概述

- ✓ 获取需求的方法
- ✓ 获取需求的过程

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 导出功能性需求

□方案只是解决问题的一个**初步轮廓**，具体的软件需求仍然需要软件工程师与利益相关方交互，听取他们的**期望和要求**

□采用多种方法导出软件需求

- ✓ 如ETC不同方案中分别有哪些软件需求？

□排除非法的软件需求，如

- ✓ 与待解决的问题无关
- ✓ 没有实际价值
- ✓ 技术不可行



3.2 构思功能性需求

- 非定制软件，很多都没有具体的客户，如微信
- 需求工程师需要充当软件利益相关方的角色，站在他们的视角，来构思软件需求
- 可以采用头脑风暴、场景分析、软件原型、群体化等多种方式来开展需求构思工作

需求工程师需要积极和主动地参与到需求获取的过程中

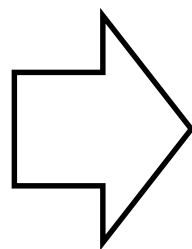
有创意软件的特点

- 创意来自于欲解决的**新颖问题**(Problems & Needs)
- 创意来自于**解决问题的独特方式**(Approach)
- 创意来自于给用户**带来的好处**(Benefits)
- 创意来自于你**强于对手**(Competitors)

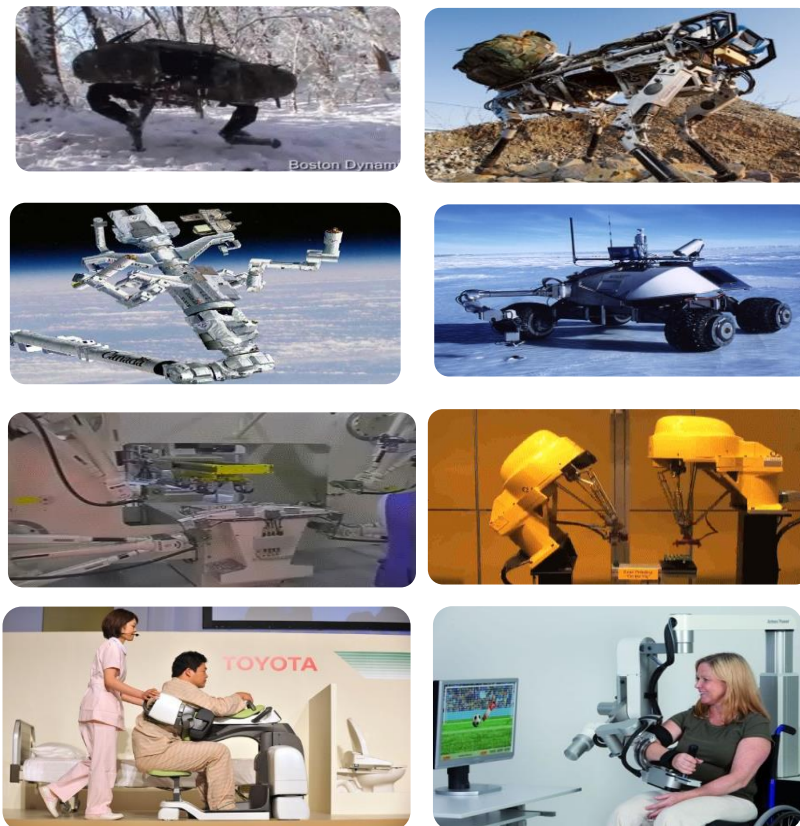
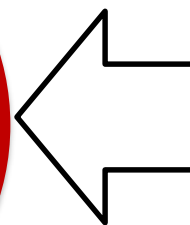
提出有创意的软件极为重要，它决定了软件的开发是否有意义和有价值

创意的潜在技术和应用领域

- 移动互联网
- 机器人
- 无人系统
- 物联网
- 可穿戴设备
- 大数据分析
-



软件作为载体和手段



潜在的技术和方法

潜在的应用和需求

软件创意示例：无人值守图书馆

要解决的问题

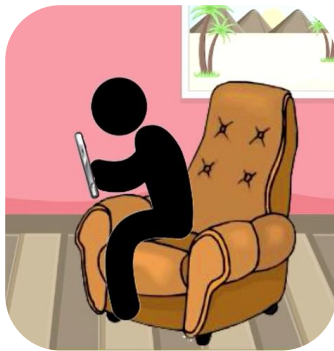
✓ 找寻图书困难

软件创意

✓ 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





国防科学技术大学

参赛队员：张元良 王腾 冯震冬 曹书锐 陈宇
指导教师：毛新军 尹良军
竞赛单位：国防科学技术大学计算机学院

无人值守的图书借阅系统

颠覆你的借阅习惯
开创图书借阅新时代

产品背景：

读者自助借书、还书、续书、借书仍然是当前图书馆实现图书借阅的主要服务模式。但是由于图书库藏的无序性、读者错乱归还造成的随意性等情况，读者和图书馆管理往往面临着需要花费大量的时间和精力去收书的问题。针对这一问题，我们将自主机器人引入到图书借阅过程，并借助移动终端的多功能计算技术，推出无人人工参与的全条件书式，将读者从繁琐的图书归还过程中解放出来。

创新价值：

- 创新图书借阅模式；
- 创新机器人与移动终端的交互；
- 机器人自主完成图书借阅任务；

实现功能：

- 读者通过移动终端发送借书请求
- 机器人接收借书任务，自主移动到图书所在地点，完成取书行为，并运送到指定的存放地点
- 机器人将取书情况及时反馈给读者，读者接收反馈信息后即可前往取书

技术路线：

- 基于Android平台开发面向读者的手机软件app
- 基于nao机器人平台开发面向取书任务的机器人驱动功能
- 基于图书馆信息系统开发用于接收和分配读者借书请求的后台系统

无人值守的图书借阅系统

机器人自主借书



基于nao机器人的借书



机器人自主还书





示例：构思“空巢老人看护软件”功能性需求

□老人视角

- ✓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

□家属和医生视角

- ✓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

站在软件利益相关方的视角来构思和导出软件需求

3.3 导出非功能性需求

□非功能性需求包括软件**质量要求**和软件**开发约束**

- ✓**质量要求**，如软件运行性能、可靠性、易用性、安全性等外部质量要求，软件可扩展性、可维护性、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

□**不满足非功能性需求**，你的产品就是一个“纸牌屋”，随时可能被“微风”击倒。



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

内容

1. 获取软件需求概述

- ✓ 获取需求的方法
- ✓ 获取需求的过程

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
- ✓ 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

- ✓目前的软件需求还很**粗略**，**不够具体和详尽**，**可能有遗漏**，**不一致和冲突等问题**，后续还需要进一步的精化和分析

□初步需求的描述方式

- ✓自然语言描述
- ✓软件原型描述
- ✓用例图描述

4.2 方式1：自然语言描述

□自然语言是最常用的需求描述手段

- ✓可为各方所理解，便于交流和讨论

□自然语言描述的局限性

- ✓**有二义**，不同的人对需求描述可能有不同的理解

- ✓**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

示例：用自然语言描述的初步软件需求

□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

- ✓客户端App软件须运行在Android 4.4及以上版本的操作系统

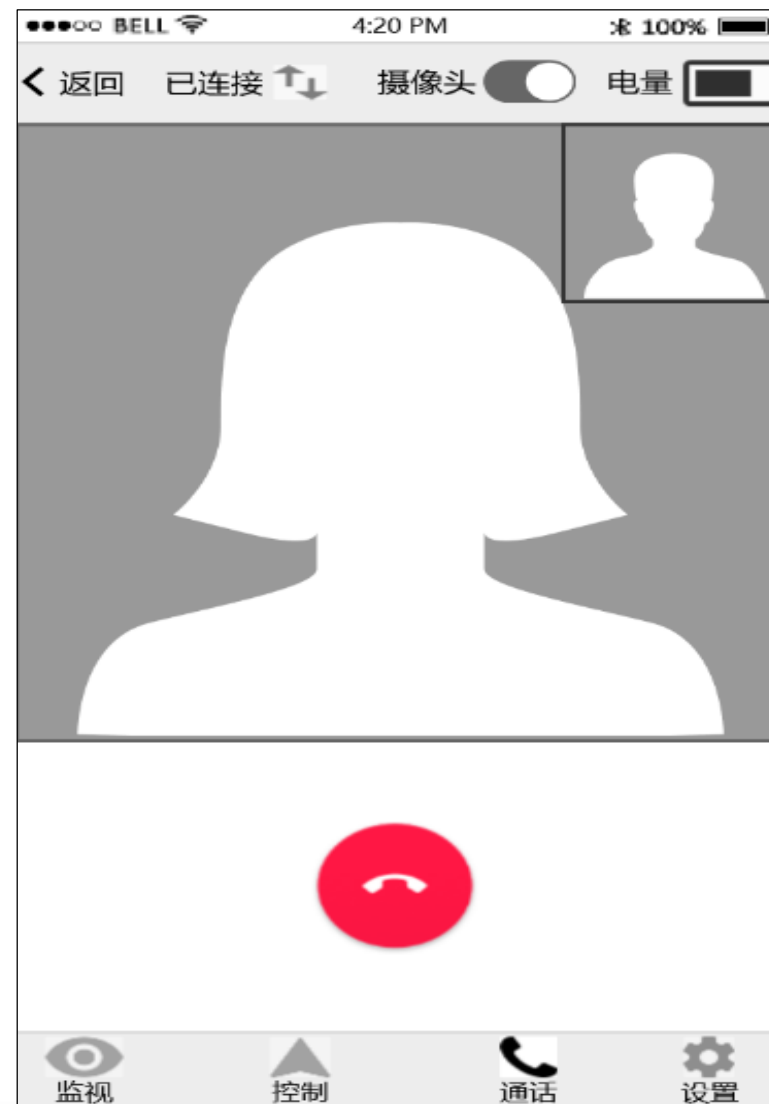
4.3 方式2：软件原型描述

□优势

- ✓ **直观、可展示和可操作**
- ✓ 通过使用软件原型，帮助用户和客户确认和导出软件需求

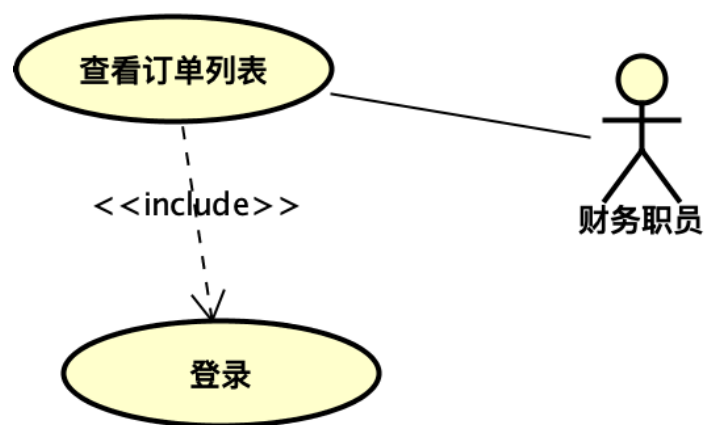
□不足

- ✓ 以操作界面的形式展示软件需求的梗概，**无法描述软件需求的具体细节**



4.4 方式3：UML用例图描述

- UML：统一建模语言
- UML 提供了一系列图形化的符号和规则，对现实世界和软件系统进行可视化建模
- 注意：UML是一种符号化语言，而不是一个开发过程



用例图



www.omg.org

UML多视点建模

□用例视点

- ✓通过**用例图**，刻画**系统的功能需求**

□结构视点

- ✓通过包图、类图、对象图和构件图，描述**系统的构成**

□行为视点

- ✓通过交互图、状态图与活动图，刻画**系统的行为**

□部署视点

- ✓通过部署图，刻画**软件制品及其运行环境**

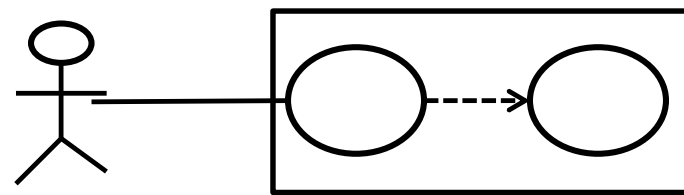
用例图

□用途

- ✓ 描述系统的**边界**，以及从**外部使用者**视角观察到的**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



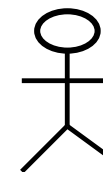
1. 执行者(Actor)

□何为执行者

- ✓ **系统外的实体**，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

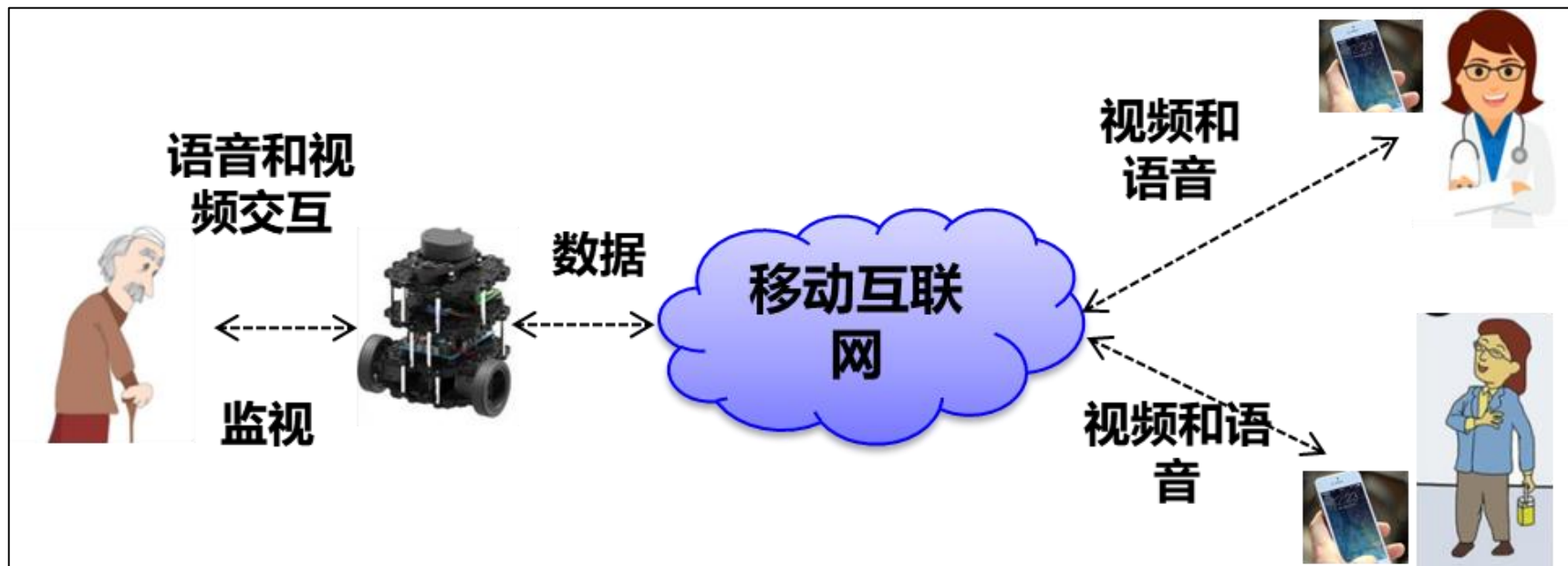
- ✓ 代表一类用户或者外部实体，而非具体的对象实例



**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

2. 用例(Use Case)

□用例

- ✓表示执行者为达成一项独立、完整的**业务目标**而要求软件系统完成的**功能**
- ✓执行者**希望系统完成的每件事**即为一个用例



Use Case

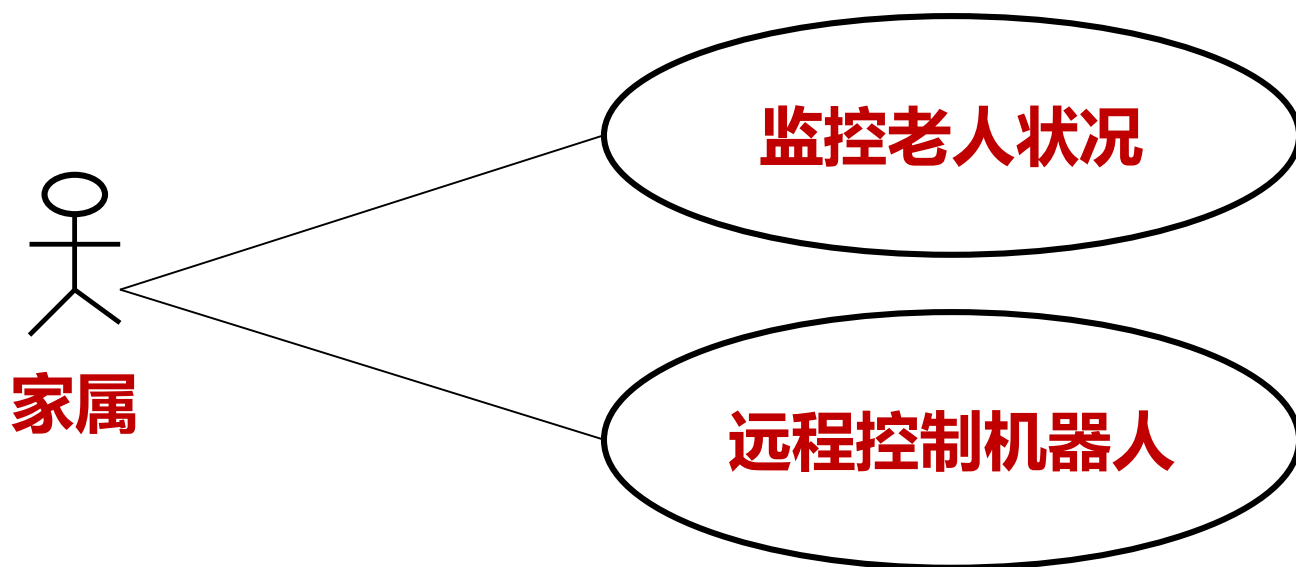
□用例的表现形式

- ✓表现为执行者与系统之间的**交互动作的序列**
- ✓对于待开发系统而言，该交互过程即是可见**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



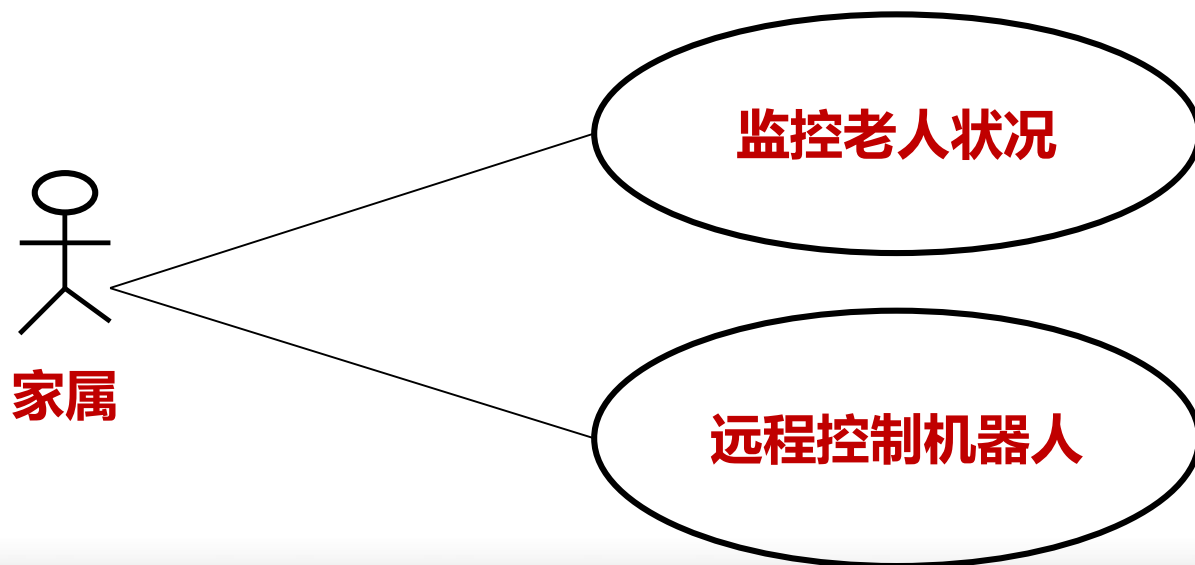
名称表示执行者

动名词表示用例

3. 执行者与用例间的关系

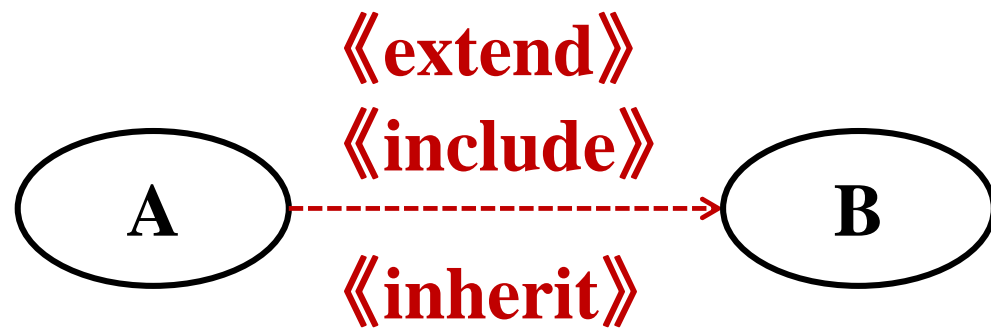
□ 执行者与用例间关系

- ✓ 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



用例间的关系

- ① 包含(Include)
- ② 扩展(Extend)
- ③ 继承(Inherit)



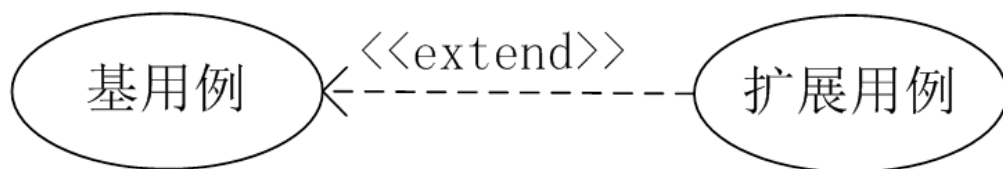
用例间的关系-包含关系

- 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
 - ✓ 包含关系用于提取多个用例中的**公共**子功能，以避免重复和冗余
 - ✓ 体现了**功能分解**和组织的思想

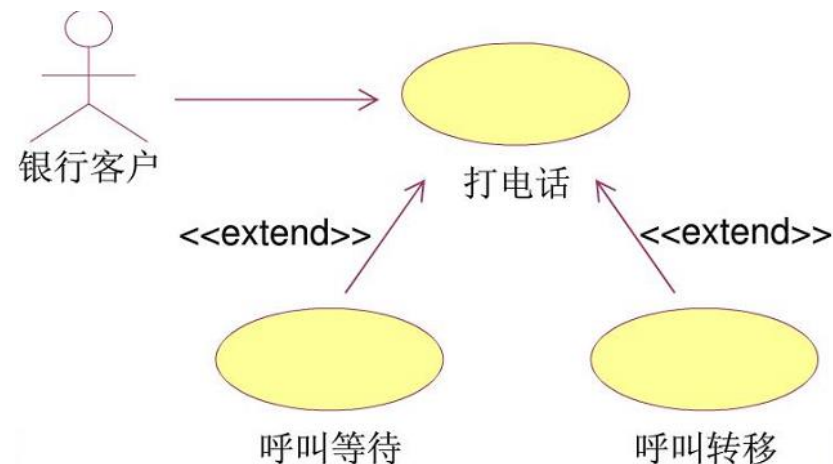


用例间的关系-扩展关系

- 表示一个用例（扩展用例）在特定条件下为另一个用例（基用例）添加新的行为或功能
- 扩展用例通常表现为例外，不会独立执行，而是依赖于基用例的执行



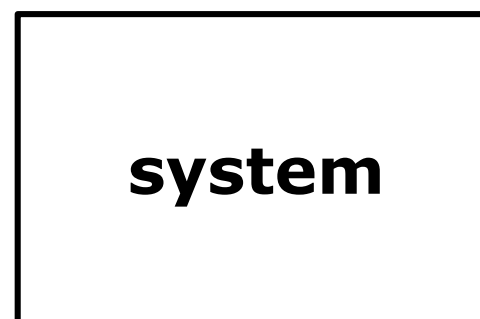
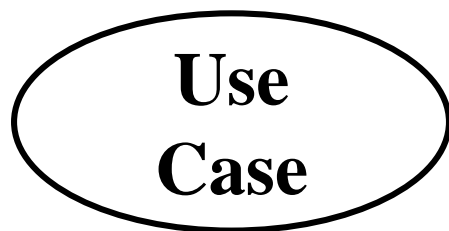
注意：扩展关系的箭头方向



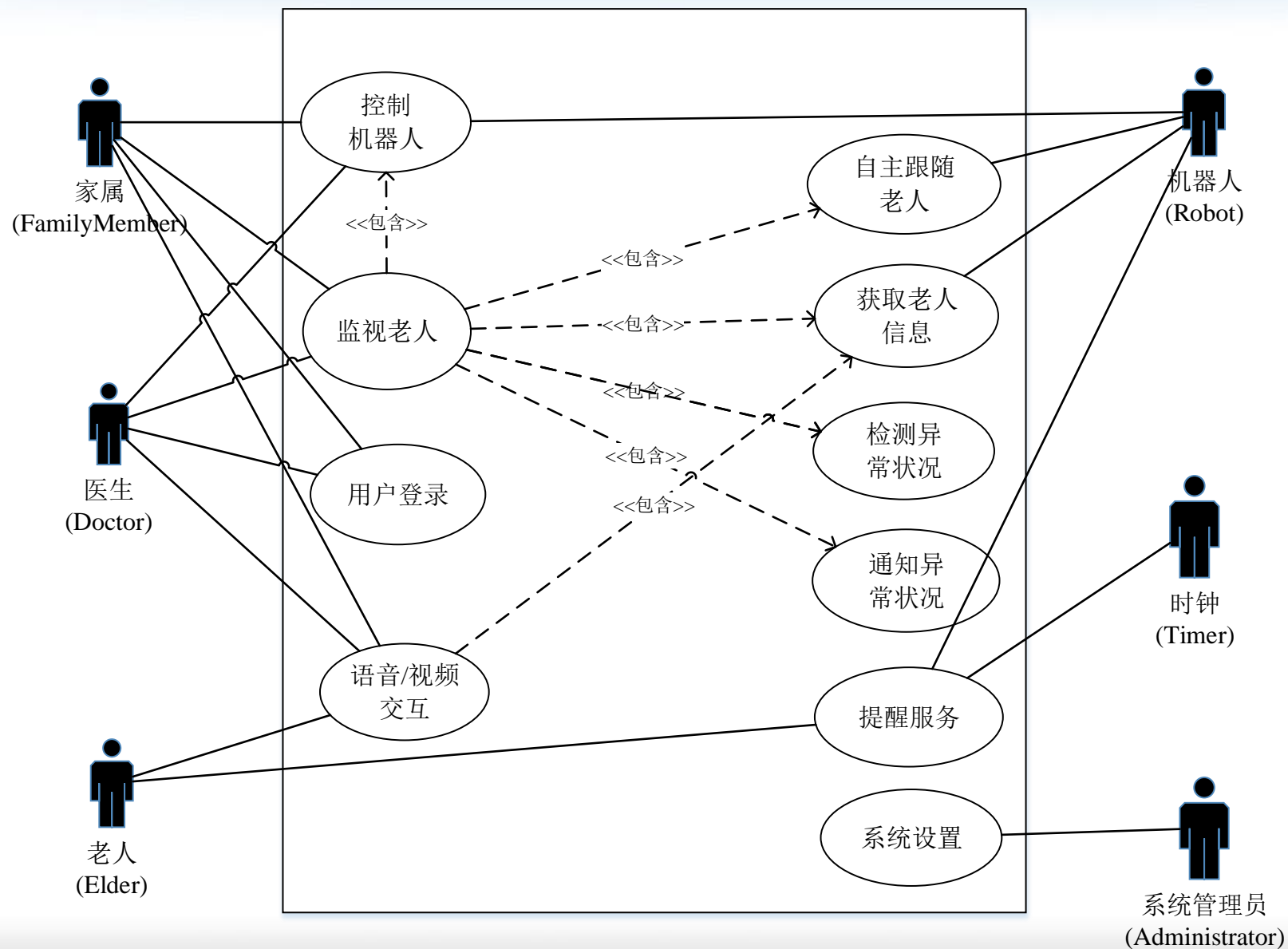
边界框

□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



示例：“空巢老人看护软件”的用例图



练习：画出“市政坑洼修复系统”的用例图

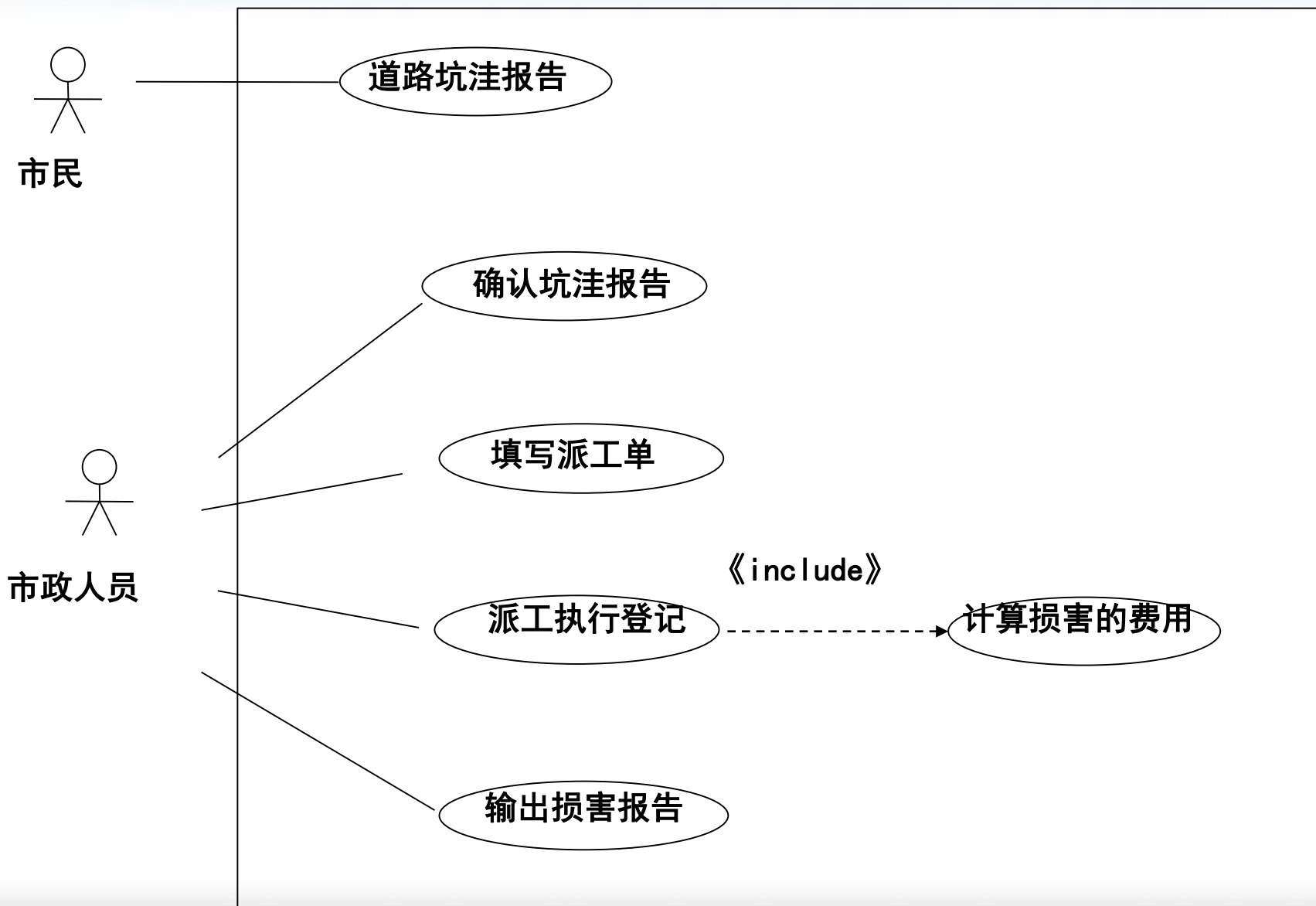
市民可以登录网站报告坑洼的位置和严重程度。每个被报告的坑洼，将登记到“市政坑洼修复系统”中，并被赋予一个标识号，按街道地址、大小、位置、城区和修复的优先级存储。

市政人员确认坑洼情况属实后，填写派工单并对派工执行情况进行登记。内容包括：位置和大小、修理队的人数、使用的修理装备、修复所用的时间、坑洼状况（正在工作、已被修理、未修理）、使用填料的数量和修理的开销（由使用的时间、人数、使用的材料的装备计算得到）。

坑洼经修复后，可输出坑洼的损害报告，包括市民的姓名、地址、电话号码、损害类型和修复总费用。

识别用例的方法是：先找系统的执行者，再考虑每个执行者对系统的目标、期望

练习：市政坑洼修复系统”的用例图



对用例的进一步详细描述

- 用例名
- 用例标识
- 执行者
- 目标
- 范围
- 前置条件
- 事件流

为什么需要进一步描述用例的这些信息？它有什么作用？

用例描述是将用例发生的各种场景描述出来，表示**执行者与系统交互时双方的行为**，即执行者做什么，系统做什么反应。

示例：用例描述

□ **用例名：** 用户登录

□ **用例标识：** UC-UserLogin

□ **主要执行者：** 家属、医生

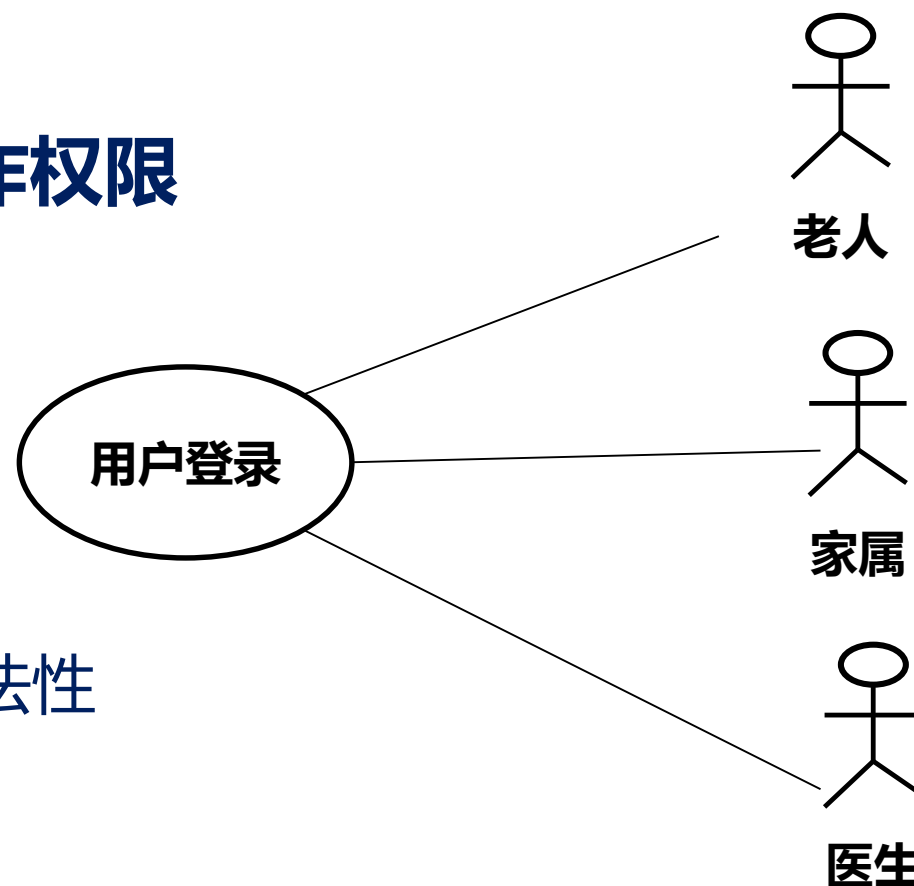
□ **目标：** 通过合法身份登录系统以获得操作权限

□ **范围：** 空巢老人看护软件

□ **前置条件：** 使用App软件之时

□ **事件流：**

- (1) 用户输入账号和密码
- (2) 系统验证用户账号和密码的正确性和合法性
- (3) 验证正确和合法则意味着登录成功
- (4) 提示登录是否成功的信息



事件流细分

□用例的事件流

- ✓说明参与者和系统之间的交互过程;

□分为主事件流和备选事件流两类

- ✓主事件流：也称基本流，描述该用例**正常执行**的一种场景；

- ✓备选事件流：描述主事件流之外的各种**可能失败情况、分支情况或扩展路径**。

示例：用例描述

用例名称	借书
参与者	图书管理员（主要参与者），读者（次要参与者）
前置条件	图书管理员已被识别和授权
后置条件	存储借书记录，更新库存数量，所借图书状态为出借
主事件流	<ol style="list-style-type: none">1. 图书管理员将读者借书卡提供给系统；2. 系统验证读者身份和借书条件；3. 图书管理员将读者所借图书输入系统；4. 系统记录借书信息，并且修改图书的状态和此种书的可借数量；
备选事件流	<ol style="list-style-type: none">2a. 非法读者<ol style="list-style-type: none">1. 系统提示读者身份错误，用例结束2b. 读者借书数已达限额<ol style="list-style-type: none">1. 系统提示读者已达结束限额，用例结束2c. 读者有过期未还书籍<ol style="list-style-type: none">1. 系统提示读者应归还的书籍列表和到期日，用例结束5a. 读者借书数已达限额<ol style="list-style-type: none">1. 系统提示，并要求结束输入2. 图书管理员确认借书完成5b. 读者有该书的预定记录<ol style="list-style-type: none">1. 删除该书的预定信息

内容

1. 获取软件需求概述

- ✓ 获取需求的方法
- ✓ 获取需求的过程

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



评审初步软件需求

□旨在发现初步需求中的问题（如不合理，遗漏），分析需求的可行性

- ✓技术可行性
- ✓经济可行性
- ✓进度可行性
- ✓设备可行性
- ✓商业可行性
- ✓社会可行性

小结

□ 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

□ 软件需求的**验证和确认**

- ✓ 多方从多个方面来评审初步的软件需求

□任务：构思开源软件的新需求

□方法

- ✓采用集体讨论的方式来构思开源软件的新需求，结合实际的问题来构思软件需求，以完善开源软件的功能和性能

□要求

- ✓所构思的软件需求要有意义和价值，存在技术可行性，具有一定的规模，需用1000行以上的代码加以实现；用自然语言和UML用例图来描述所构思的软件需求，撰写相应的软件需求文档

□结果：UML用例图模型和软件需求描述文档

□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

问题和讨论

