

**物品检测与抓取机器人**  
**需求规格说明书**  
***SRS102***  
**V2.0.0**

### 分工说明

小组名称	rushbot	
学号	姓名	本文档中主要承担的工作内容
17373060	杨开元	合作完成业务需求与功能需求部分
17373517	王正达	合作完成业务需求与功能需求部分
15241035	彭文鼎	完成其它需求与开发环境部分
17373288	尹俊成	完成项目概述与数据需求部分

### 版本变更历史

版本	提交日期	主要编制人	审核人	版本说明
V1.0.0	2020.3.30	尹俊成	彭文鼎	本小组需求规格说明初稿
V2.0.0	2020.4.4	尹俊成	彭文鼎	本小组需求规格说明二稿

## 目录

1. 范围.....	1
1.1 项目概述.....	1
1.2 文档概述.....	2
1.3 术语和缩略词.....	2
1.4 引用文档.....	3
2. 业务需求.....	3
2.1 总体描述.....	3
2.2 用例图.....	4
2.3 用例详细描述.....	4
3. 数据需求.....	9
3.1 指令类.....	9
3.2 目标检测、路径规划类.....	9
3.3 运动控制类.....	10
3.4 地图类.....	10
3.5 特征学习类.....	10
3.6 异常类.....	10
4. 功能需求.....	11
4.1 功能一(用例三).....	11
4.2 功能二(用例四).....	12
4.3 功能三(用例五).....	13
4.4 功能四(用例六).....	14
5. 非功能需求.....	15
5.1 系统性能需求.....	15
5.2 系统质量需求.....	15
6. 用户界面需求.....	16
7. 运行与开发环境.....	17
7.1 硬件环境.....	17

7.2 软件环境.....	17
---------------	----

# 1. 范围

## 1.1 项目概述

### 1.1.1 开发背景

随着计算机技术的发展，网络的普及，硬件和制造业的进步，我们在越来越多的领域都能看到机器人的身影。人类希望将那些枯燥重复、有流程约定的任务交到价格低廉、可以全天候工作的机器人手中去，进而提高效率，减少成本。

近些年来，机器人领域不断在取得进步，机器人开发也离普通人越来越近。ROS 系统就是一个适用于机器人开发的开源操作系统，提供了操作系统应有的服务，如抽象、消息传递、包管理等等。ROS 还是一个分布式的节点框架，进程被封装在易于分享和发布的程序包和功能包中，工程也可以被 ROS 的基础工具方便地整合在一起。

与此同时，越来越多的新技术也正在 ROS 系统中发挥作用，如开源的 OpenCV、PCL 图像库、语音识别，结合这些技术机器人也有了更广泛的作用领域，更好的人机交互体验等等。

### 1.1.2 主要功能和非功能性需求

我们项目的最终目标是实现一个可以自主移动、主动避障以及对目标物进行检测和抓取的简易机器人。它拥有自主分析地图信息以及在此基础上预先进行路径规划的功能。为改善人机交互体验，我们也会尝试着加入语音识别与控制系统。

在功能性需求之外，我们还需要保证机器人能够稳定地完成高质量的工作，即系统的可靠性。此外，系统的可扩展性需求也很重要。机器人系统应能够适应一定程度内系统容量的扩大和管理内容的增加。最后，为了改善用户的体验以及扩大机器人的适用人群，机器人系统使用起来应该足够简单。

### 1.1.3 应用场景

该机器人简单易用，功能较为全面的特点使得其有很多值得展望的应用场景。

该机器人可以在无人职守的商店里 24 小时值班并销售物品，面对提前规格化的货架和机器人自己建立的地图，只需要让其学习每种商品的特征，加上语音识别模块，就能很好地和人类交互，让这种机器人在夜间或全天服务再好不过。

通过优化分类算法和大量数据的学习，改进机械臂的构造，该机器人可以实现物品的分拣，可以被安排在物流公司这种急需劳动力、垃圾分类点这种环境较差的场景中。

此外，该机器人也可以在家中充当保姆的角色，将其安置在行动不便老人的家中，通过老人和机器人的互动，机器人可以学习用户身边的物品，并在用户发出指令之后将物品取来或归位。

## 1.2 文档概述

本文档用于对项目的需求与业务逻辑进行细致化地分析。依次介绍了项目的业务需求、数据需求、功能需求、非功能性需求以及项目的基本开发环境。

## 1.3 术语和缩略词

表 1 术语和缩略词列表

术语和缩略词	简要解释
ROS	Robot Operating System 编写机器人软件的高度灵活性框架
OpenCV	Open Source Computer Vision Library 轻量高效的开源计算机视觉库
PCL	Point Cloud Library 实现 3D 视图下处理计算的点云库

## 1.4 引用文档

- [1] 王鹏飞. 基于 ROS 的多传感器信息融合自主导航控制系统设计与实现[D].南京邮电大学,2019.
- [2] 北京六部工坊科技有限公司. 启智 ROS 机器人开发手册 V1.1.0

## 2. 业务需求

### 2.1 总体描述

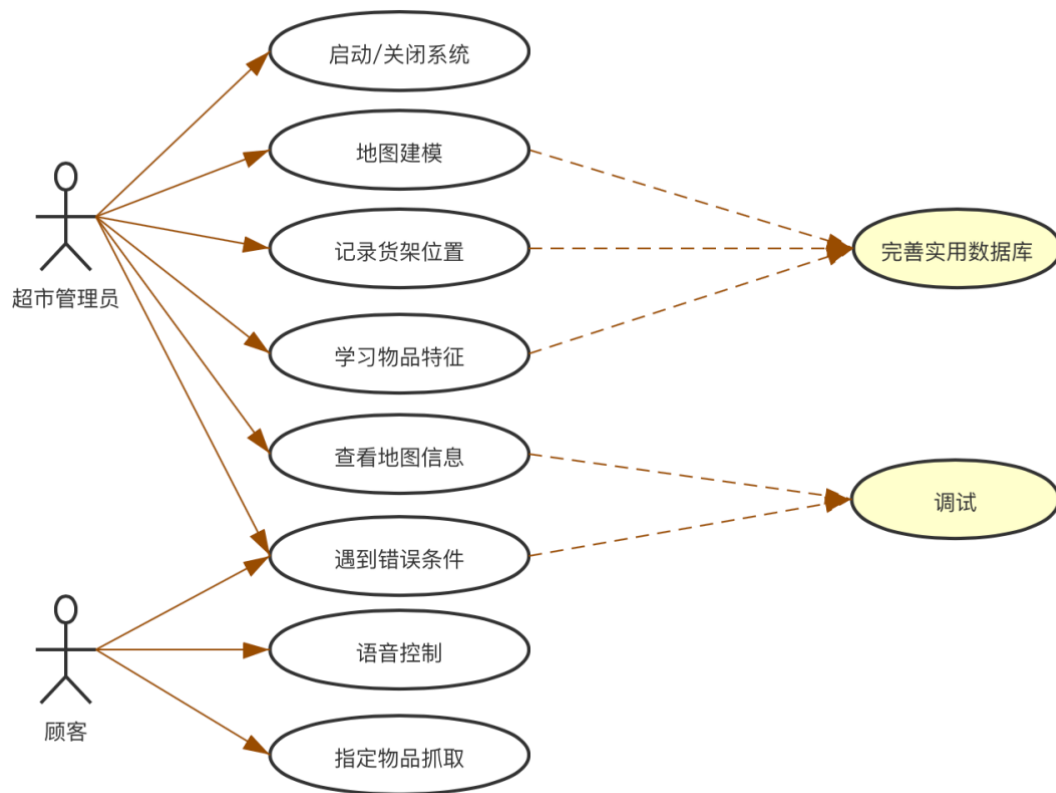
机器人主要使用场景为在空间有限、货架有限的超市内。主要任务是协助用户购物，无人售货等。

在顾客使用之前，该机器人需要超市管理员对机器人进行相关配置。首先机器人将在管理员带领下进行地图建模。然后，超市管理员将带领机器人行走并记忆货架位置。当机器人行进到特定货架位置时，需要超市管理员对当前位置标定，并输入当前货架上的商品种类。并且对于需要机器人进行识别并抓取的物品，机器人需要进行特征学习。以上工作完成之后机器人可以在超市入口或特定位置等待顾客。

顾客进入超市时，对机器人发出语音指令，说明待机器人抓取的物品，机器人将当前位置记录为顾客位置，之后机器人从当前位置自动规划路径前往包含该物品的货架，进行物品的识别和抓取。最后，机器人导航回到初始位置，发出相应提示。在整个业务流程中，机器人实现自主避障。完成一个业务循环。

超市管理员可以通过机器人机载电脑或者远程控制的方式启动、关闭机器人；通过 UI 界面查看地图及机器人所处位置；查看机器人对物品特征的学习情况；查看完成业务次数等等调试控制信息。机器人将对成功获取到的指令及时做出相应，在到达取物地点和抓取物品成功后发出提示，并对错误和异常情况做出判断、处理和提示，以提供良好的可调试性。

## 2.2 用例图



## 2.3 用例详细描述

### 2.3.1 用例一：启动机器人

**主要参与者：**超市工作人员

**目标：**使机器人系统各模块进入工作状态，打开人机交互界面

**前置条件：**机器人各模块配置完毕；超市工作人员了解规范启动步骤

**启动：**超市工作人员希望使用机器人

**场景：**

1. 接通机器人电源；
2. 通过 USB 接口连接机载电脑和机器人运动控件；
3. 启动机载电脑并打开机器人系统；
4. 机载电脑显示人机交互界面；

**优先级：**高



**何时可用：**第一个增量

**使用频率：**低

**次要参与者：**机器人运动控件、机载电脑

### 2.3.2 用例二：关闭机器人

**主要参与者：**超市工作人员

**目标：**关闭机器人，确保下次启动时各模块状态正常

**前置条件：**机器人各模块配置完毕；超市工作人员了解规范关闭步骤；机器人未处于工作状态

**启动：**超市暂时不再需要机器人

**场景：**

1. 关闭机器人系统；
2. 解除机载电脑和机器人运动控件的 USB 连接；
3. 关闭机载电脑；

**优先级：**高

**何时可用：**第一个增量

**使用频率：**低

**次要参与者：**机器人运动控件、机载电脑

### 2.3.3 用例三：初始化地图场景建模

**主要参与者：**超市工作人员

**目标：**确保机器人系统正确记录超市地图

**前置条件：**机器人系统已正常启动；超市工作人员熟悉超市地形

**启动：**超市工作人员点击“开始建图”按钮

**场景：**

1. 机器人放置于超市入口处并提示启动成功；
2. 超市工作人员点击“开始建图”按钮；
3. 超市工作人员手动操控机器人在超市内穿梭移动；
4. 机载电脑人机界面实时显示地图建模情况；
5. 超市工作人员操控机器人回到超市入口；
6. 超市工作人员通过机载电脑界面查看场景建模情况；

7. 确保建模无误后，超市工作人员点击“停止建图”按钮；
8. 机器人收到指令，回到待命状态；

**异常情况：**

1. 场景建模不完善 出现点：6

**解决方案：**

1. 工作人员点击“停止建图”按钮，操作机器人回到超市入口，点击“开始建图”按钮，重新开始地图建模；

**优先级：** 高

**何时可用：** 第一个增量

**使用频率：** 中

**次要参与者：** 机器人运动控件、机载电脑

#### **2.3.4 用例四：货架地点记忆**

**主要参与者：** 超市工作人员

**目标：** 确保机器人系统正确记录货架位置

**前置条件：** 机器人系统已正常启动；超市工作人员熟悉超市地形；所有商品名称已录入系统数据库；地图建模已经完成；

**启动：** 超市工作人员点击“记忆货架”按钮

**场景：**

1. 机器人放置于超市入口处并提示启动成功；
2. 超市工作人员点击“记忆货架”按钮；
3. 超市工作人员手动操控机器人在超市内移动，前往某一货架；
4. 机器人到达货架前，超市工作人员点击“记录位置”按钮，并选择相应一  
众商品名称；
5. 机器人收到指令，记录商品位置并在地图上标注；
6. 超市工作人员检查该商品位置记录情况；
7. 重复 3-5 步，直至所有商品位置记录完毕；
8. 超市工作人员操控机器人回到超市入口；
9. 超市工作人员检查所有商品位置记录情况；
10. 超市工作人员点击“停止记忆”按钮；

11. 机器人进入待命状态；

**异常情况：**

1. 某商品位置记录不准确 出现点：6
2. 某商品位置未被记录 出现点：9

**解决方案：**

1. 超市工作人员小范围内移动机器人进行位置校准，校准之后重新点击“记录位置”按钮并选择相应商品名称；
2. 超市工作人员操作机器人重新前往被遗漏商品位置，进行记录；

**优先级：**中

**何时可用：**第二个增量

**使用频率：**中

**次要参与者：**机器人运动控件、机载电脑

### 2.3.5 用例五：商品特征学习

**主要参与者：**超市工作人员

**目标：**使机器人具备识别区分目标物品的能力

**前置条件：**机器人处于待命状态；

**启动：**超市工作人员点击“开始学习”按钮并输入当前商品名称

**场景：**

1. 机器人处于待命状态
2. 机器人放置于货架前，需满足光线充足，高度适当
3. 超市工作人员点击“开始学习”按钮并输入当前商品名称，机器人启动计时器
4. 将商品合理摆放在货架上
5. 机器人记录商品特征
6. 若完成学习，机器人显示“已完成学习”，转 7  
    若计时器时间超过 5 分钟，机器人显示“学习超时”，转 7  
    若未完成学习，显示“请转动物品以展示其他角度”，转 4
7. 机器人回到待命状态

**异常情况：**

1. 步骤 4、5 中，机器人多次学习仍未完成学习

**解决方案：**

1. 为应对异常情况 1，超市工作人员应调整物品所处环境高度、光线环境

### 2.3.6 用例六：指定商品获取

**主要参与者：** 顾客

**目标：** 命令机器人到相应货架取回指定商品

**前置条件：** 超市地图初始化已完成，指定商品位置已正确记录；机器人处于待命状态；

**启动：** 顾客点击“获取商品”按钮并选择目标商品；

**场景：**

1. 机器人放置于超市入口处并处于待命状态；
2. 顾客在机器人交互界面上点击“获取商品”按钮并选择目标商品；
3. 机器人收到指令，记录顾客所在位置，提示“马上出发，请您在原地耐心等待”；
4. 机器人规划前往指定商品所在货架的路径；
5. 机器人移动，路上进行自主避障；
6. 机器人到达目标货架；
7. 机器人摄像头识别商品并确定位置；
8. 机械臂启动，开始抓取目标商品；
9. 目标商品抓取成功；
10. 机器人规划返回路径；
11. 机器人移动，路上进行自主避障；
12. 机器人返回顾客所在位置；
13. 机器人松开机械臂，将商品移交给顾客；
14. 机器人进入待命状态；

**异常情况：**

1. 机器人在移动过程中遇到无法绕过的障碍物
2. 机器人返回顾客位置时顾客已经离开

**解决方案：**

1. 机器人停在原地，发出警报声，并显示“请帮我移开障碍物”提醒周围的超市管理人员前来处理，管理人员移开障碍物之后，机器人继续移动。

2. 机器人停在原地，发出警报声，并显示“我找不到顾客了”，提醒周围的超市管理人员前来处理，管理人员前来取走商品放回货架，并停止机器人的报警状态。

### 3. 数据需求

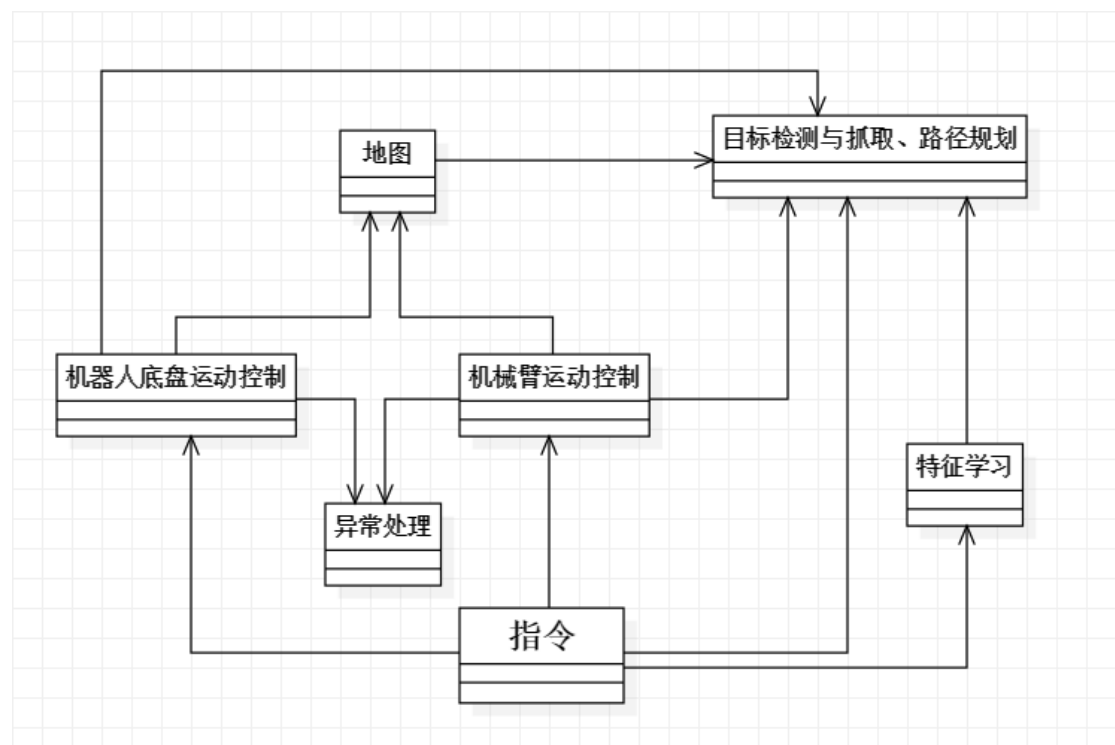


图 1 类层次结构图

#### 3.1 指令类

接收客户的输入并且控制机器人完成相应的任务。如运动控制、特征学习、抓取指定物品等。

#### 3.2 目标检测、路径规划类

在其它类的辅助下完成目标检测以及路径规划等工作。

### **3.3 运动控制类**

完成对机器人的运动控制工作，分为底盘运动控制和机械臂运动控制。

### **3.4 地图类**

完成与地图构建相关的工作。

### **3.5 特征学习类**

完成与目标物的特征学习相关的工作。

### **3.6 异常类**

处理机器人服务过程中可能出现的错误情况，如识别不出所指定的物品等。

## 4. 功能需求

### 4.1 功能一(用例三)

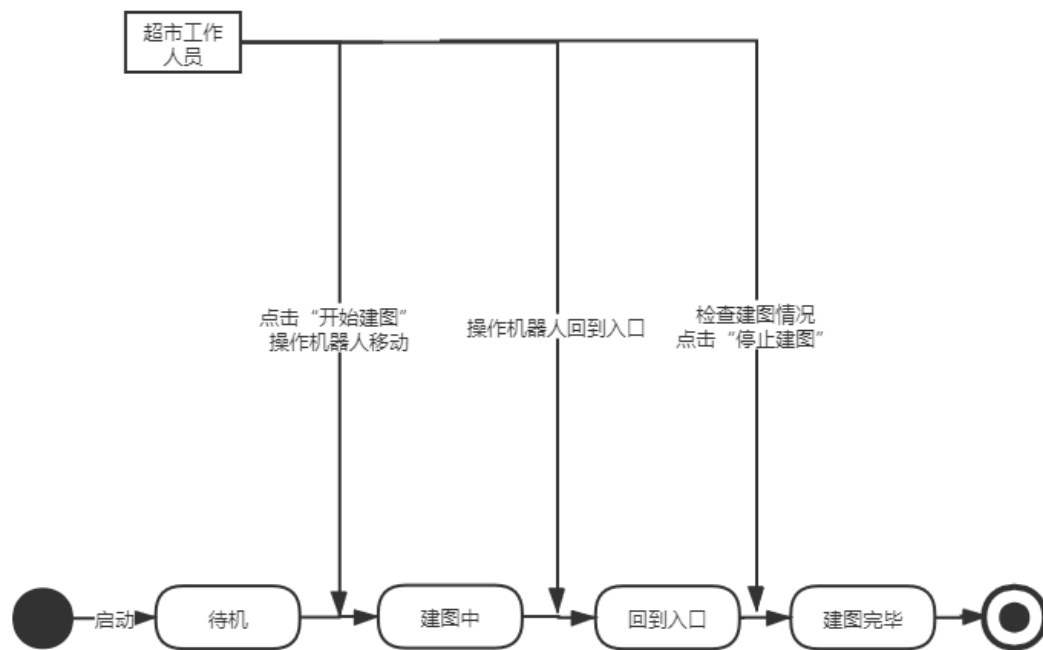


图 2 用例三示意图

## 4.2 功能二(用例四)

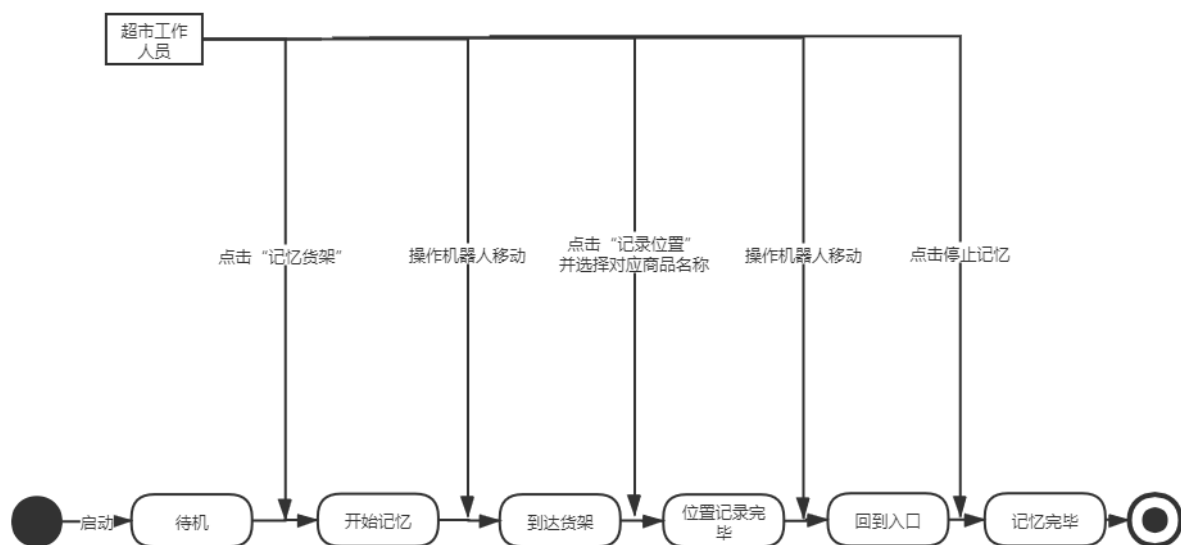


图 3 用例四示意图



### 4.3 功能三(用例五)

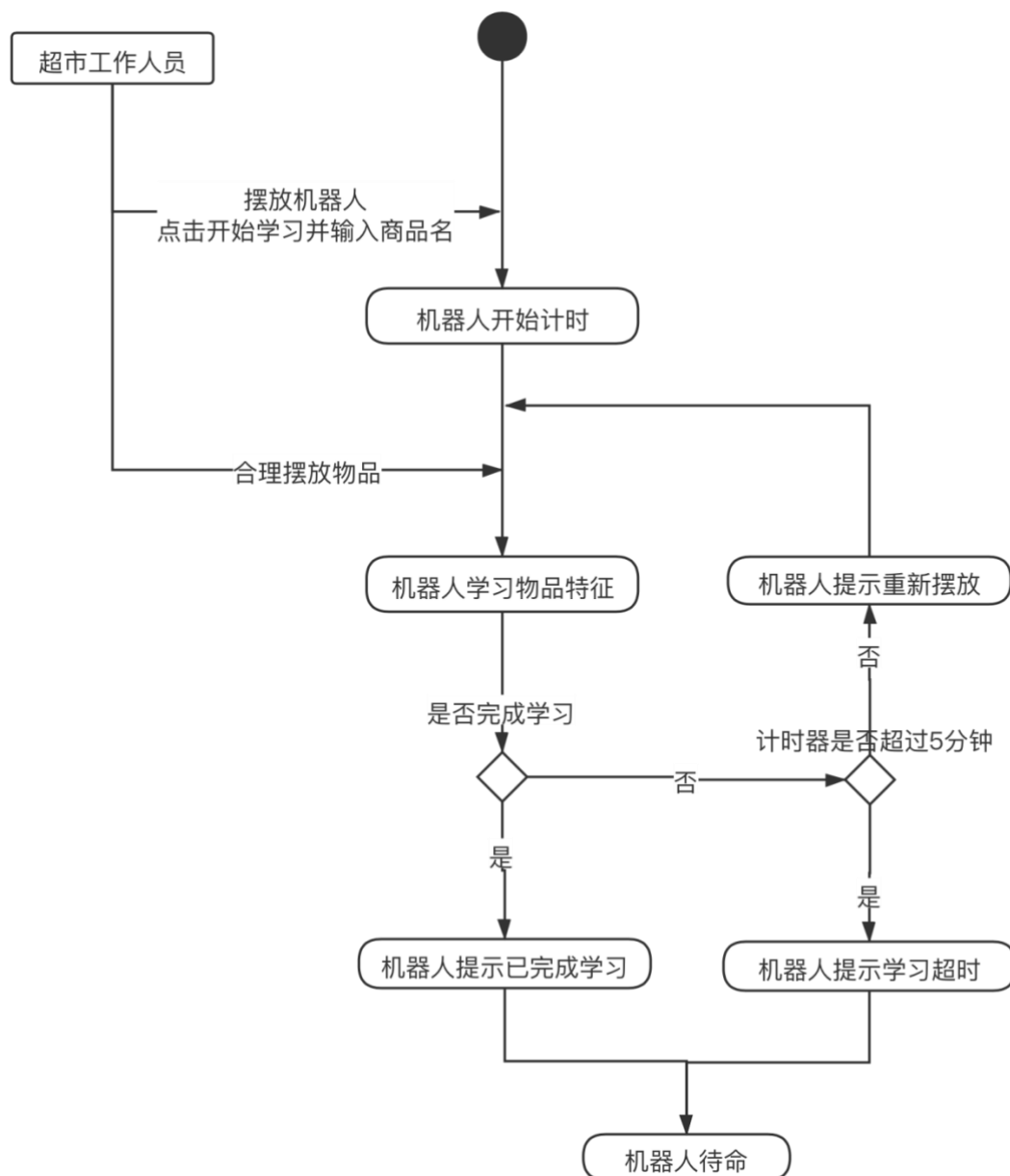


图 4 用例五示意图

#### 4.4 功能四(用例六)

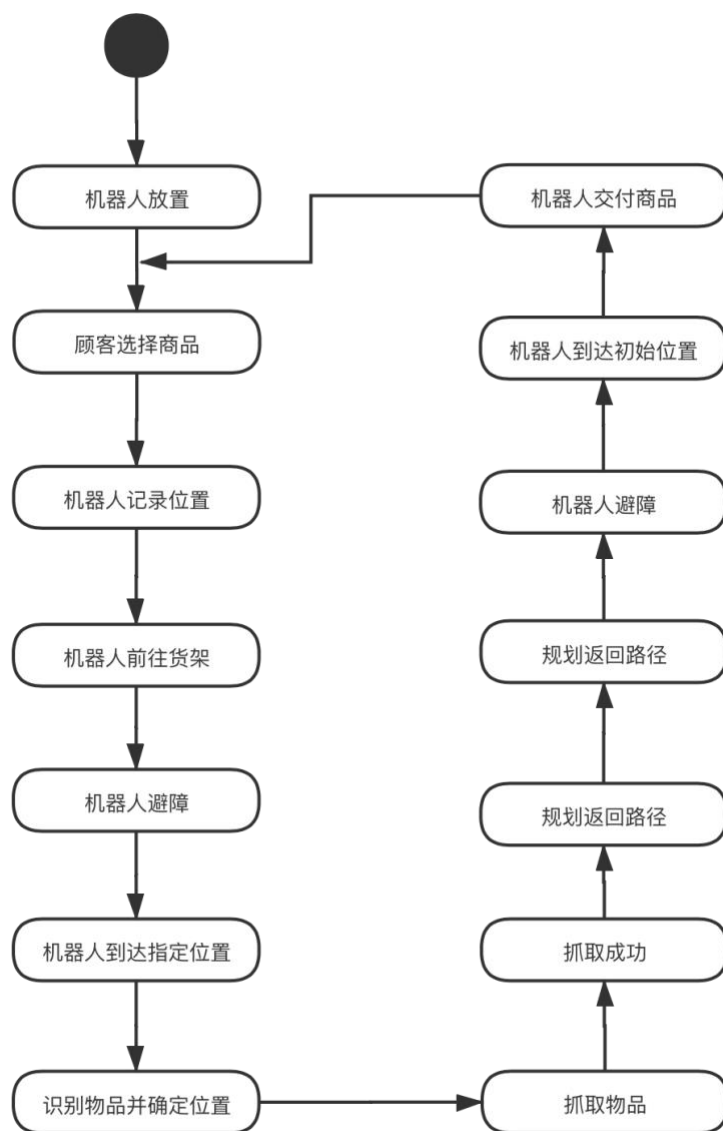


图 5 用例六示意图

## 5. 非功能需求

### 5.1 系统性能需求

#### 5.1.1 响应时间

机器人系统的各个功能环节都要求较短的响应时间，尤其是取货时需要较高效率，否则会大大降低用户体验和业务表现。同时客户端界面跳转需要实时，比如不超过 3s。

#### 5.1.2 资源利用率

指系统投入服务器、数据库、硬件设备等资源，所发挥的资源利用百分比。希望投入的资源最大化的利用，而不被闲置。系统不必要且消耗资源大的数据或功能需要定期清理或优化。

### 5.2 系统质量需求

#### 5.2.1 安全性

- **保密性：**数据加密保护，保证数据在采集、传输和处理过程中不被偷窥、窃取、篡改。业务数据需要在存储时进行加密，确保不可破解。
- **防泄漏：**通过对文档进行读写控制、打印控制、剪切板控制、拖拽、拷屏截屏控制、和内存窃取控制等技术，防止泄漏机密数据。
- **权限控制：**根据用户权限控制访问数据，进行操作记录等等。
- **防攻击：**IP 限制、高频访问限制等等，如：用户高频点击，有时不是恶意，但也有可能造成系统异常。

#### 5.2.2 可维护性与可扩展性

机器人设备应具备良好的可维护性和可扩展性，能够适应系统内容的扩大和管理内容的增加，包括软硬件平台、系统结构、功能设计。随着功能数量增加和

复杂化，要求系统具有能够便于持续维护和灵活扩展。

### 5.2.3 可靠性

系统要求设备避免单点失效的容灾能力，同时还需要具有软硬件故障在线恢复的能力，需要调试、应急处理的机制以便配置和管理。

### 5.2.4 易用性

系统对用户而言需要易于学习和使用，设计的重点在于让产品的设计能够符合使用者的习惯与需求。具体需求包括易学习性、易操作性、用户错误防御机制、用户界面美观等。

## 6. 用户界面需求

顾客通过语音发送指令控制机器人交互，用户界面则是系统的控制面板。

用户界面提供的信息有：一是向用户反馈机器人响应结果，二是为用户提供指令输入模块，三是可以查看机器人状态以及任务完成情况等信息。

超市管理员可以通过机器人机载电脑或者远程控制的方式启动、关闭机器人；通过图形界面查看地图及机器人所处位置；查看机器人对物品特征的学习情况；查看完成业务次数等等调试控制信息。

## 7. 运行与开发环境

### 7.1 硬件环境

运行和开发的硬件环境基本相同：

嵌入式开发板+各类传感器（激光雷达、RGB-D 摄像头等等）+运动装置+机械臂装置

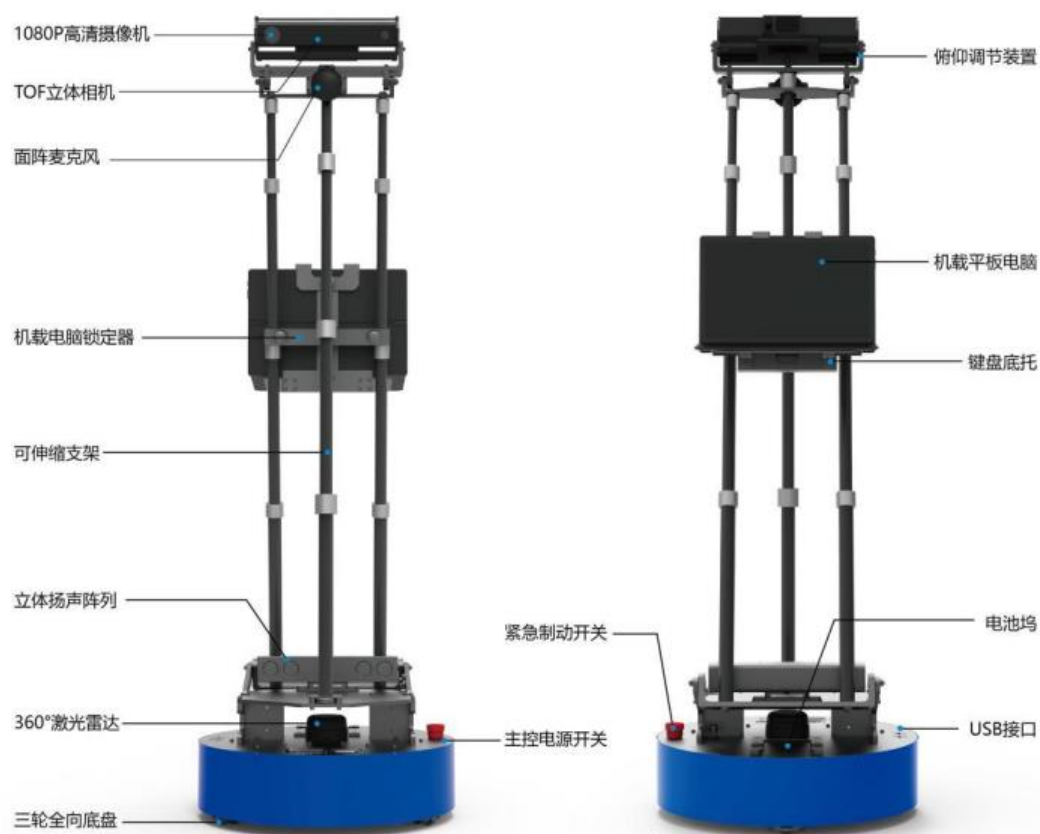


图 6 项目所用机器人示意图

### 7.2 软件环境

运行和开发环境：

Ubuntu 16.04

ROS Kinetic

此外，开发时的 IDE 使用 Roboware Studio。

需要的软件包：

1. 启智机器人基础软件包

表 2 启智机器人软件包

Package 名称	内容
wpb_home_bringup	启智 ROS 机器人的基础功能
wpb_home_behaviors	启智 ROS 机器人的行为服务
wpb_home_tutorials	启智 ROS 机器人的应用例程
wpbh_local_planner	启智 ROS 机器人的导航局部规划器

2. 扩展软件包

启智机器人除了基础功能的软件包以外，还提供了一个扩展软件包，该软件包里包含了大量复合任务的实现例程。因为该软件包需要依赖 xfyun\_waterplus 和 waterplus\_map\_tools 等第三方软件包，所以和基础软件包相互独立，以方便持续维护更新。