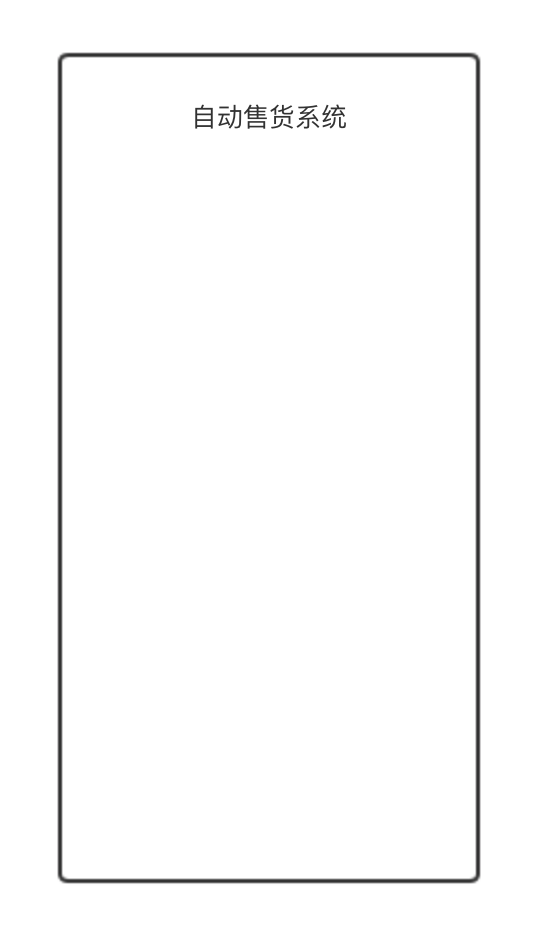


一幅**用例图**包含的模型元素有系统、行为者、用例及用例之间的关系。图是自动售货机系统的用例图。图中的方框代表系统，椭圆代表用例（售货、供货和取货款是自动售货机系统的典型用例），线条人代表行为者，它们之间的连线表示关系。

**1.系统**

**系统**被看作是一个提供用例的黑盒子，内部如何工作、用例如何实现对于建立用例模型来说都是不重要的。

代表**系统**的方框的边线表示系统的边界，用于划定系统的功能范围，定义了系统所具有的功能。描述该系统功能的用例置于方框内，代表外部实体的行为者置于方框外。



**2.用例**

一个**用例**是可以被行为者感受到的、系统的一个完整的功能。在UML中把**用例**定义成系统完成的一系列动作，动作的结果能被特定的行为者察觉到。这些动作除了完成系统内部的计算与工作外，还包括与一些行为者的通信。用例通过**关联**与行为者连接，**关联**指出一个用例与哪些行为者交互，这种交互是双向的。

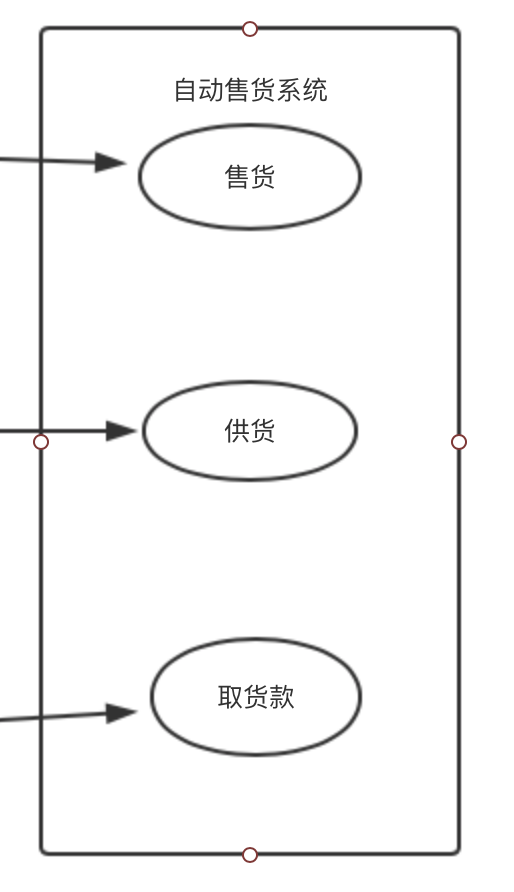
用例具有下述特征。

(1)用例代表某些用户可见的功能，实现一个具体的用户目标。

(2)用例总是被行为者启动的，并向行为者提供可识别的值。

(3)用例必须是完整的。

**注意**，用例是一个类，它代表一类功能而不是使用该功能的某个具体实例。用例的实例是系统的一种实际使用方法，通常把用例的实例称为**脚本**。脚本是系统的一次具体执行过程。



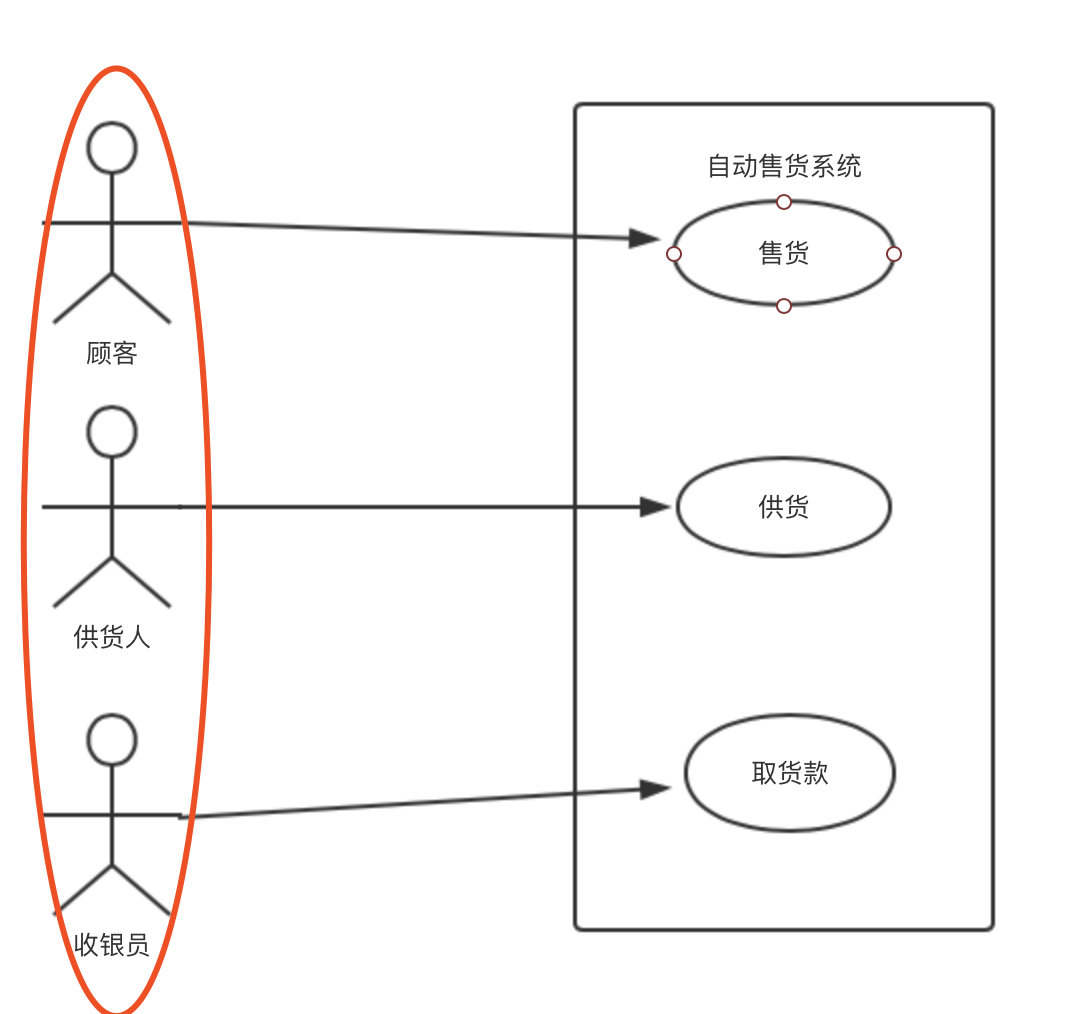
**3.行为者**

**行为者**是指与系统交互的人或其他系统，它代表外部实体。使用用例并且与系统交互的任何人或物都是行为者。

**行为者**代表一种角色，而不是某个具体的人或物。一个具体的人可以充当多种不同角色。

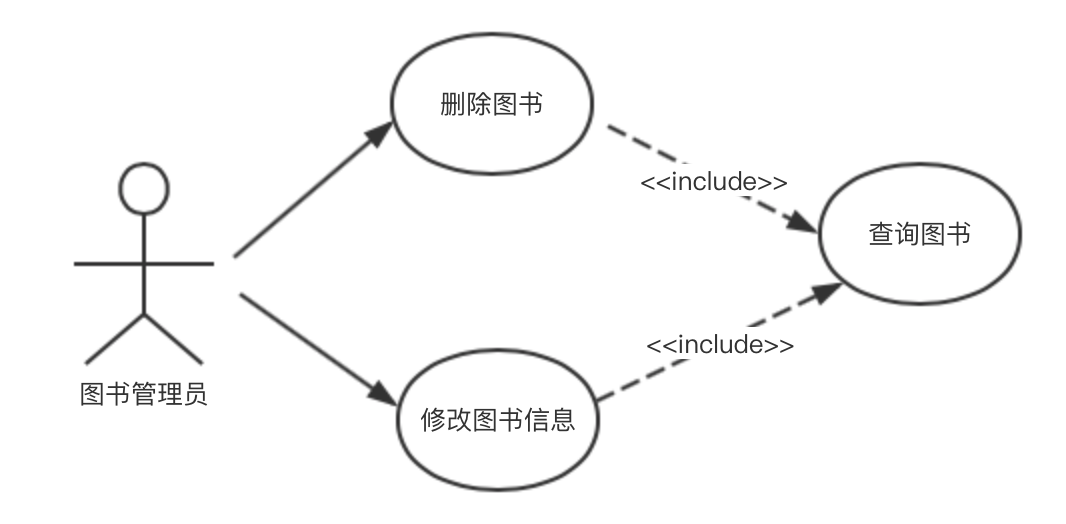
在用例图中用直线连接行为者和用例，表示两者之间交换信息，称为**通信联系**。行为者触发(激活)用例，并与用例交换信息。单个行为者可与多个用例联系；一个用例也可与多个行为者联系。

可以把行为者分成主行为者和副行为者，还可分成主动行为者和被动行为者。

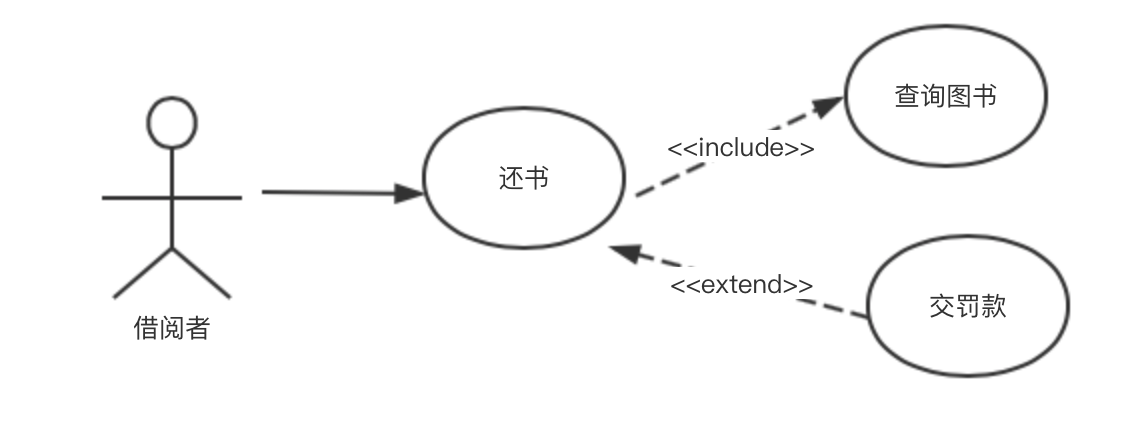


**4.用例之间的关系**

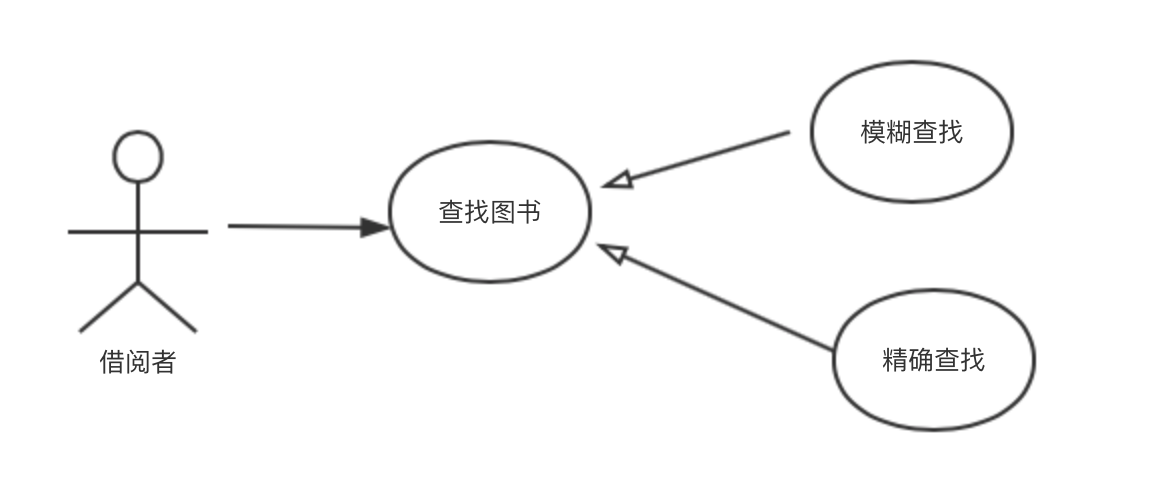
[**包含关系**](https://baike.baidu.com/item/%E5%8C%85%E5%90%AB%E5%85%B3%E7%B3%BB)：基本[用例](https://baike.baidu.com/item/%E7%94%A8%E4%BE%8B)的行为包含了另一个用例的行为。基本用例描述在多个用例中都有的公共行为。包含关系本质上是比较特殊的依赖关系。它比一般的依赖关系多了一些语义。在包含关系中箭头的方向是从基本用例到包含用例。



**扩展关系:** 基本含义和泛化关系类似，但在扩展、关系中，对于扩展用例有更多的规则限制，基本用例必须声明扩展点，而扩展用例只能在扩展点上增加新的行为和含义。与包含关系一样，扩展关系也是依赖关系的版型。在扩展关系中，箭头的方向是从扩展[用例](https://baike.baidu.com/item/%E7%94%A8%E4%BE%8B)到基本用例，这与包含关系是不同的。



**泛化关系**：代表一般与特殊的关系。它的意思和[面向对象程序设计](https://baike.baidu.com/item/%E9%9D%A2%E5%90%91%E5%AF%B9%E8%B1%A1%E7%A8%8B%E5%BA%8F%E8%AE%BE%E8%AE%A1)中的继承的概念是类似的。不同的是继承使用在实施阶段，泛化使用在分析、设计阶段。在泛化关系中子[用例](https://baike.baidu.com/item/%E7%94%A8%E4%BE%8B)继承了父用例的行为和含义，子用例也可以增加新的行为和含义或者覆盖父用例中的行为和含义。



# 创建用例图模型

一个**用例模型**由若干幅用例图组成。创建用例模型的工作包括：**识别出系统中的角色和用例**；**区分用例之间的先后关系**；**创建用例图模型结构**。其中，识别出系统中的角色和用例是关键。

## 1. 识别出系统中的角色和用例

为获取用例首先要找出系统的行为者，可通过请系统的用户回答一些问题来发现行为者。下述问题有助于发现行为者。

* 谁将使用系统的主要功能（主行为者）？
* 谁需要借助系统的支持来完成日常工作？
* 谁来维护和管理系统（副行为者）？
* 系统控制哪些硬件设备？
* 系统需要与哪些其他系统交互？
* 哪些人或系统对本系统产生的结果（值）感兴趣？

一旦找到了行为者，就可以通过请每个行为者回答下述问题来获取用例。

* 行为者需要系统提供哪些功能？行为者自身需要做什么？
* 行为者是否需要读取、创建、删除、修改或存储系统中的某类信息？
* 系统中发生的事件需要通知行为者吗？行为者需要通知系统某些事情吗？从功能观点看，这些事件能做什么？
* 行为者的日常工作是否因为系统的新功能而被简化或提高了效率？

还有一些不是针对具体行为者而是针对整个系统的问题，也能帮助建模者发现用例，例如：

* 系统需要哪些输入输出？输入来自何处？输出到哪里去？
* 当前使用的系统（可能是人工系统）存在的主要问题是什么？

## 2. 区分用例之间的先后关系

区分用例优先次序某些用例必须在其他用例之前完成，因为它们之间要相互依赖。例如，在系统借阅图书之前，必须记录图书的基本信息。因此很明显新增图书是最重要的用例。

### 3．创建用例图模型结构

将已确定并细化的角色和用例放人用例图中。此时，再借助包含、扩展和泛化的关系给出用例之间的结构模型。

在系统需求分析中需考虑系统用例图模型需要哪些视图，每个视图包含什么内容?视图中成员是否需构成包? 下面针对\_述的图书管理系统·为其建立系统的用例图模型。

图书管理系统整体用例图（不知道该不该放）

