

本文由 [简悦 SimpRead](https://www.simpread.com/) 转码，原文地址 www.imooc.com

《手册》在设计规约部分对用例图有这样一条规定：

【强制】在需求分析阶段，如果与系统交互的 User 超过一类并且相关的 User Case 超过 5 个，使用用例图来表达更清晰的结构化需求。

用例图是需求分析的一种重要图形工具，本小节我们将学习用例图的概念，用例图的核心组件，用例图的使用场景和使用案例等。

2.1 什么是 UML？为什么要使用 UML？

UML 即统一建模语言，是一种用于说明、可视化、构建和编写面向对象、软件密集型系统的开放方法。UML 对大规模、复杂系统的建模有极大的帮助。

UML 通过它的元模型和表示法，把通过文字和其他表达方法难以表达清楚的内容，简单直观的通过图形表达出来。

使用 UML 图可以让我们和客户，让软件开发的各个角色之间的沟通交流更加顺畅。实际开发中，我们画出各种 UML 图，前端、测试甚至产品都可以很容易地通过 UML 快速看明白我们的设计方案，这就是 UML 的价值所在。

就 Java 开发工程师而言，UML 图通常出现在技术文档中。通过 UML 图来表达我们的系统设计，帮助其他成员理解和评估我们的方案。另外这些需求梳理或者技术方案，对后面维护的同学有极大的参考价值。

可能会有很多同学会说：**我平时不画图也不影响开发啊？画图完全是在浪费时间。**

有这种认识的核心原因在于抵触心理，不敢离开舒适区；另外一点在于参与的项目都比较小，无法体会需求分析、系统复杂时 UML 图体现出来的价值。

很多朋友，尤其是新手，在需求分析阶段和方案设计阶段不重视，往往导致后期设计偏移需求，遗漏需求，甚至设计方案重新设计，部分代码重新编写等情况，反而浪费更多时间，付出更大的代价。

UML 图形主要分为结构型图形，行为型图形两大类。UML 2.2 包含 14 中图，分别如下：



本专栏重点介绍工作中常用的：用例图、类图、时序图、状态机图和活动图。

2.2 用例图是什么？为什么要使用用例图？

用例图是一种以**用户视角**来描述**系统功能**的图形。

用例图包括：系统（System）、用例（Use Case）、参与者（Actor）以及用例和参与者之间的关系。

其中被描述的事物就是系统；系统的参与者称为角色；角色在系统中要做的的事就是用例也叫行为。

用例图通常用在**需求分析**和**总体设计**阶段。**用例图的目的**是让项目的参与者能够在更高层次上理解系统。通过用例图可以基于用户视角对大型项目的功能进行拆分，而**任务分解又是降低复杂度的核心方法之一**。

2.3 核心组件

2.3.1 基本组件

如下图所示，参与者使用小人图标表示，系统通过方形表示，用例一般采用椭圆形表示，参与者和用例之间通过连线来表示关系。

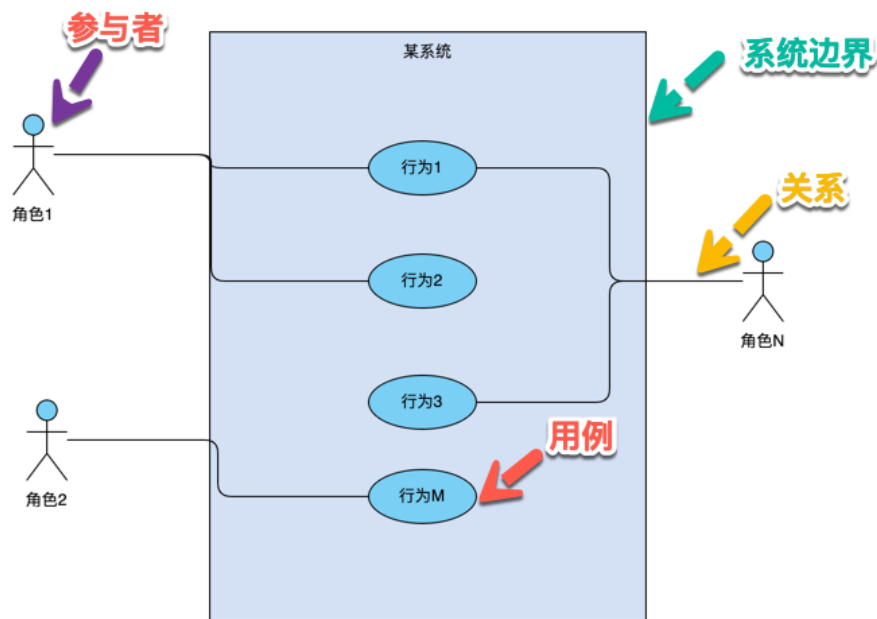


图 1：用例图的基本示例图

2.3.2 关系描述

用例图中的关系主要是参与者和用例的关系，参与者和参与者的关系，用例和用例的关系为主。

其中参与者和用例的关系比较常见，如上图所示，角色 2 和行为 M 通过连线来表示他们之间的关系，即角色 2 使用该系统的目的之一或者该系统给角色 2 提供的功能之一就是行为 M。

参与者和参与者之间可能是并列的关系也可能是泛化关系，即面向对象语言中的继承关系。一般用户、管理员、超级管理员之间可以有继承关系。

用例和用例之间主要有 3 种关系，一种是包含关系（include），一种是拓展关系（extend），还有一种是继承关系。

其中**包含关系**描述一个用例需要某种功能，而该功能被定义为另外一个用例，使用带有 "<<Include>>" 的虚线箭头表示。如果管理学生信息，包括新增学生信息，修改学生信息，查询学生信息，删除学生信息。

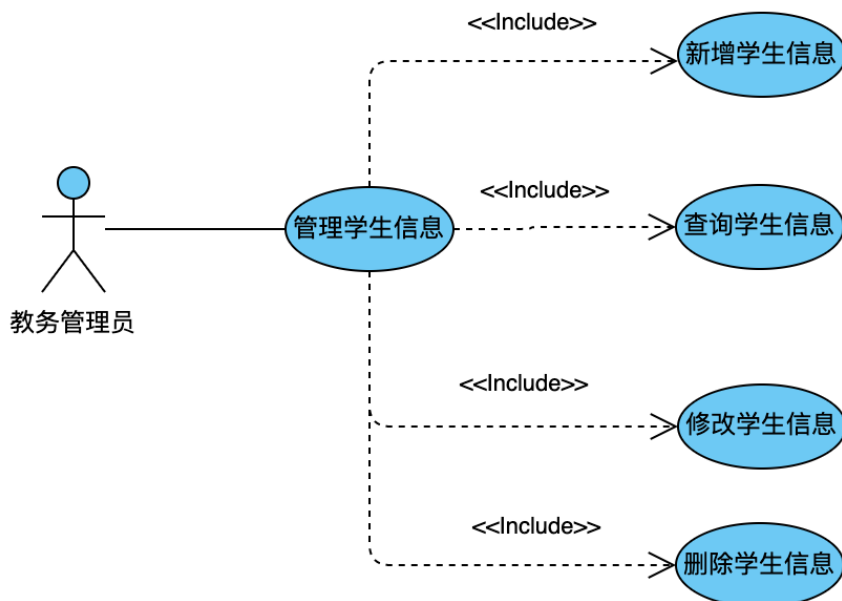


图 2：用例图包含关系示例

而**拓展关系**通常表示在某个用例的基础上，还能做什么事情，使用带有 "<>" 的虚线箭头表示。如下图所示，在读新闻的基础上，还支持打印新闻和分享新闻到朋友圈。

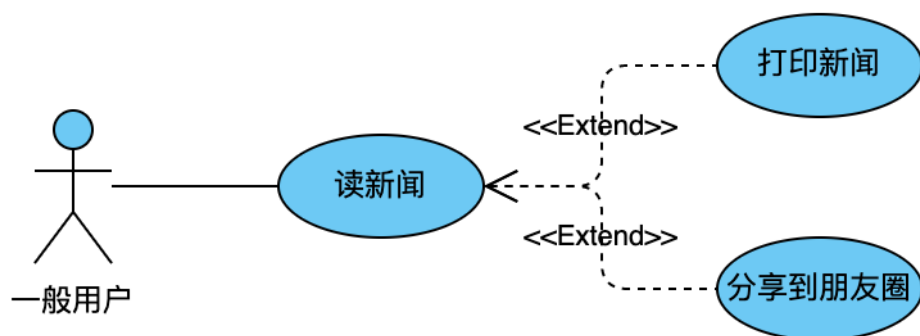


图 3：用例图拓展示例

大家可以回想一下 Java 中的继承，子类可以继承父类的属性和行为，且自己可以自定义自己的特有行为。而**用例的继承**与之类似。如下图所示，电视上投放广告具备一般投放广告行为，又有自己的特殊性，如支持视频；虽然广播的广告投放也属于投放广告的行为，但是通过音频方式传播；报纸的广告投放同样具备广告投放的功能，但是一般通过图片和文字来呈现。再如上一节讲到的登录，可以分为用户名密码登录和手机号短信登录，他们之间也是继承关系。

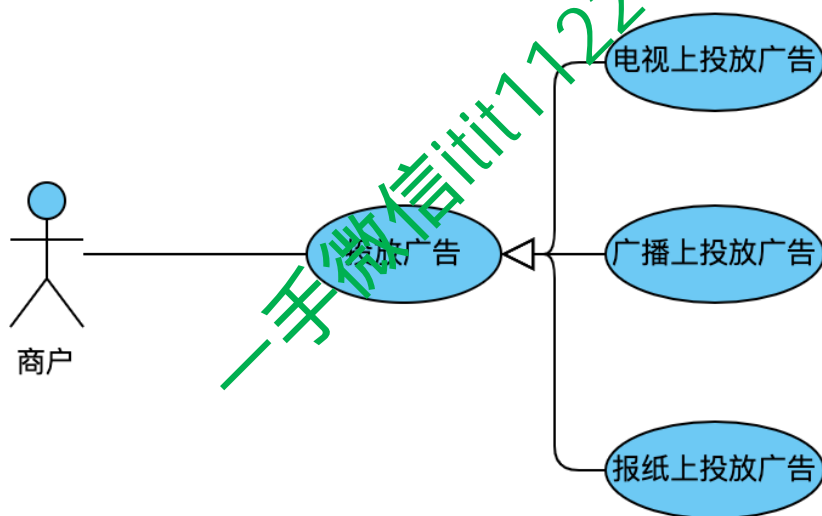


图 4：用例图的继承示例

3.1 画图工具

还是那句话：“工欲善其事必先利其器”。

那么画各种 UML 图有哪些不错的工具呢？

在学习和工作中接触了很多画流程图、UML 图工具，其中比较有名的有 PlantUML、Microsoft Office Visio、Visual Paradigm、StarUML、Process On、亿图图示等。

我将这些画图软件分为两类，一类是文本语法类，一类是拖拽类。

PlantUML 有自己的语法，它是一种“文本型”，按照规定的语法可以快速作图，功能强大，支持时序图、用例图、类图、活动图、组件图、状态图、对象图、部署图和定时图等。PlantUML 的**主要优点**是：作图快速，文本存储方便修改，用户只需要关心内容，不需要过度关心样式，同时这也是它的**缺点**是输出的图片风格较为单一。

而其他软件则拖拽为主，优点是发挥的余地很大，缺点是作图效率相对较低，尤其是对于有些“强迫症”的小伙伴来说，简直是灾难，经常需要花费很多时间在各种内部组件和连线的对齐上。

因此如果无特殊喜好，画 UML 图形 个人最推荐 PlantUML。如果你喜欢拖拽，windows 系统用户建议使用 Visio 来画图，Mac 系统用户建议使用 Visual Paradigm。

本小节使用 Visual Paradigm 进行绘图。

3.2 准备

3.2.1 寻找参与者

想要画用例图，一个重要前置环节就是寻找参与者，参与者不仅包括人还可能包括系统。

首先提出几个问题：

- 系统给哪些人设计的？由哪些人使用？
- 系统由谁来负责维护？
- 系统由谁来管理？
- 系统为哪些人或系统提供数据？

这些问题都可以帮助我们快速找到用例。

如我们分析自动售卖机系统的用例：该系统是买东西给消费者，是设计给消费者用的；售卖机需要定期补货就需要货品管理人员；售卖机可能损坏，就需要更换或维修，因此需要运维人员的参与；售卖机的屏幕可以显示广告，因此可能吸引广告商来投放广告。

3.2.2 确定用例

画用例图要明确有哪些用例，我们可以借助一系列提问来帮助我们明确用例：

- 参与者为什么要使用这个系统？
- 参与者是否有增删改查数据的行为？增删改查数据是谁来操作的？

另外我们还可以借助用例的特征来帮助我们判断用例是否正确：

用例之间相互独立。即每个用例不需要与其他用例交互就可以满足参与者的一个目的。用例从功能上是完整的，用例本质上体现了系统参与者的愿望，如果不能完整表达参与者的愿望就不能称为用例。如去 ATM 机取钱，查询余额、取款等可以成为用例，而插入银行卡就不能称之为用例，因为它是其他用例的一个前置条件，而不是一个完整功能，也不是用户参与 ATM 机系统的目的。

用例的执行结果对参与者来说是可观测和有意义的。即使有些功能时系统不可或缺的一部分，但是对用户而言是不可观测的，在需求分析阶段也不应该作为用例出现。另外作为一个单独事件对用户而言是无意义的也不应该成为用例，比如上节讲到的用户登录功能，那么登录可以称之为一个用例，而填写用户名、填写密码等就不能称之为用例，因为用户的核心目的是登录，单纯地填写用户名或者填写密码对用户而言是无意义，它们只是实现登录功能的一个重要步骤。

用例必须由参与者发起，不应该自动启动，也不应该没有参与者，更不应该主动启动另外一个用例。这也为我们寻找用例提供了一个突破口，当我们找全参与者后，可以围绕参与者的目的来推出用例。如商品售卖系统的用例包括消费者，消费则使用该系统的目的是购买商品；用例还包括运维人员，他们主要负责贩卖机的维修；以此类推，可以将相关的用例都推出来。

用例必然是动宾结构。即用例是由动作和目标构成。如取款、转账、注册账户、退出系统等。

3.2.3 确定关系

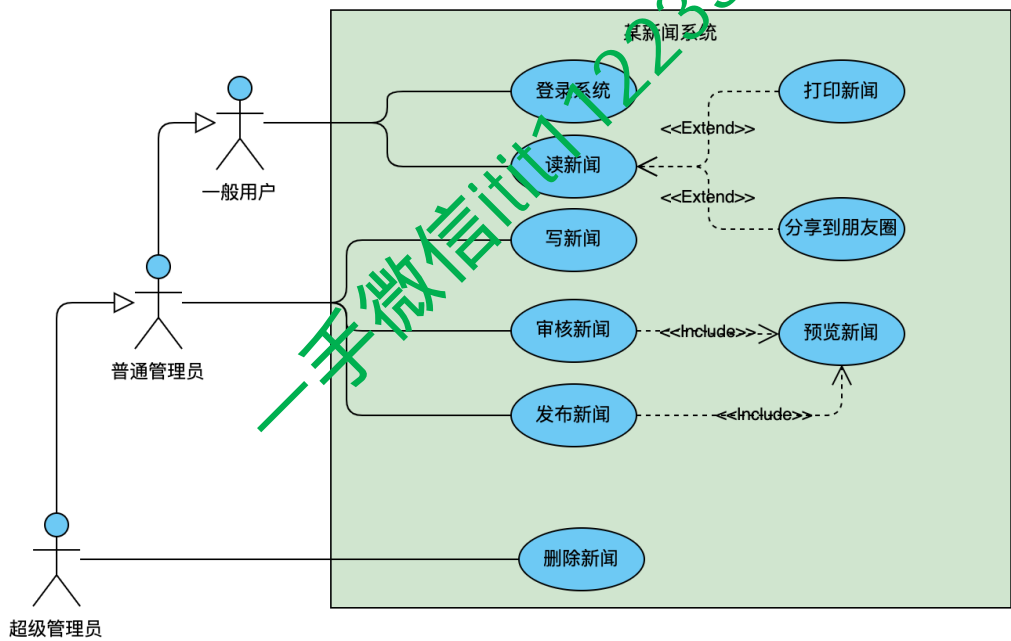
确定好参与者和用例后，就要思考参与者之间，参与者和用例之间，用例和用例之间的关系。
梳理好这些关系以后，在作图时使用 2.3 所提供的图形绘制对应的关系。

2.3 作图

某产品给出了一个需求描述：

- 我们要设计一个新闻系统，这个系统为了满足 xxxx 的需求，在这个系统中普通用户可以登录，读新闻；管理员除了具备一般用户的功能外，还可以写新闻、审核新闻，发布新闻；高级管理员具有以上所有功能外，还可以删除新闻。
- 普通用户不仅可以读新闻，将新闻内容打印并可以支持分享到朋友圈，满足用户的个性化需求，增强用户的体验，提高互动度等。
- 管理员在审核新闻和发布新闻之前要可以预览新闻，避免审核和发布失误。
- 高级管理员在删除新闻之前要有弹窗提醒。

从上面的描述我们可知系统就是某新闻系统；参与者就有三类：普通用户、普通管理员、超级管理员；核心用例有登录、读新闻、写新闻、审核新闻、发布新闻、删除新闻等。
根据以上的分析，我们可以大致画图如下：



图：某新闻系统用例图

画完此用例图后，我们重新对比需求，进行校对。如果发现该有却没有的功能，要分析是隐含在需求背后的功能，还是产品遗漏掉的功能，如果是重要的产品遗漏的功能，可以和产品反馈。比如我们发现退出系统的功能虽然产品没有给出，但是应该要有的功能，我们需要后续补上去。

在此要特别提醒大家，这需求分析时发现产品设计可能不合理的地方，可能遗漏的地方要及时沟通反馈。很多产品后期新增的需求，都是早期遗漏的需求。如果我们能够帮助产品在需求分析阶段找出遗漏的需求，就尽可能的避免后期被加需求。我们要做一个会思考的程序员，而不是执行命令的机器。

本节主要讲述了用例图的概念、目的，用例图的主要组成部分和对应图形，并给出了用例图的用法示例。希望大家可以掌握用例图，通过用例图从用户的视角来分析系统。更多高级用法推荐通过《大象：Thiking in UML》、《火球：UML 大战需求分析》相关章节深入学习。

下一节我们将学习类图，了解类图的概念、为什么要使用类图、类图的基本用法等。

相信大家都有去 ATM 机取钱的经历，请大家分析如果我们设计 ATM 系统，使用用例图进行需求分析，参与者有哪些？有哪些用例？自己动手画一画。

}

一手微信itit1223344