

系统分析师软件过程概述：用例建模指南 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/141/2021\\_2022\\_\\_E7\\_BB\\_E7\\_BB\\_9F\\_E5\\_88\\_86\\_E6\\_c29\\_141454.htm](https://www.100test.com/kao_ti2020/141/2021_2022__E7_BB_E7_BB_9F_E5_88_86_E6_c29_141454.htm)

用例(Use Case)是一种描述系统需求的方法，使用用例的方法来描述系统需求的过程就是用例建模。用例方法最早是由Iva Jackboson博士提出的，后来被综合到UML规范之中，成为一种标准化的需求表述体系。用例的使用在RUP中被推崇备至，整个RUP流程都被称作是"用例驱动"(Use-Case Driven)的，各种类型的开发活动包括项目管理、分析设计、测试、实现等都是以系统用例为主要输入工件，用例模型奠定了整个系统软件开发的基础。

1. 什么是用例？在介始用例方法之前，我们首先来看一下传统的需求表述方式-"软件需求规约"(Software Requirement Specification)。传统的软件需求规约基本上采用的是功能分解的方式来描述系统功能，在这种表述方式中，系统功能被分解到各个系统功能模块中，我们通过描述细分的系统模块的功能来达到描述整个系统功能的目的。采用这种方法来描述系统需求，非常容易混淆需求和设计的界限，这样的表述实际上已经包含了部分的设计在内。由此常常导致这样的迷惑：系统需求应该详细到何种程度？一个极端就是需求可以详细到概要设计，因为这样的需求表述既包含了外部需求也包含了内部设计。在有些公司的开发流程中，这种需求被称为"内部需求"，而对应于用户的原始要求则被称之为"外部需求"。功能分解方法的另一个缺点是这种方法分割了各项系统功能的应用环境，从各项功能项入手，你很难了解到这些功能项是如何相互关联来实现一个完成的系统服务的。所以在传

统的SRS文档中，我们往往需要另外一些章节来描述系统的整体结构及各部分之间的相互关联，这些内容使得SRS需求更象是一个设计文档。

### 1.1 参与者和用例

从用户的角度来看，他们并不想了解系统的内部结构和设计，他们所关心的是系统所能提供的服务，也就是被开发出来的系统将是如何被使用的，这就用例方法的基本思想。用例模型主要由以下模型元素构成：

- 参与者(Actor)**参与者是指存在于被定义系统外部并与该系统发生交互的人或其他系统，他们代表的是系统的使用者或使用环境。
- 用例(Use Case)**用例用于表示系统所提供的服务，它定义了系统是如何被参与者所使用的，它描述的是参与者为了使用系统所提供的某一完整功能而与系统之间发生的一段对话。
- 通讯关联(Communication Association)**通讯关联用于表示参与者和用例之间的对应关系，它表示参与者使用了系统中的哪些服务（用例），或者说系统所提供的服务（用例）是被哪些参与者所使用的。这大三种模型元素在UML中的表述如下图所示。

以银行自动提款机(ATM)为例，它的主要功能可以由下面的用例图来表示。ATM的主要使用者是银行客户，客户主要使用自动提款机来进行银行帐户的查询、提款和转帐交易。通讯关联表示的是参与者和用例之间的关系，箭头表示在这一关系中哪一方是对话的主动发起者，箭头所指方是对话的被动接受者；如果你不想强调对话中的主动与被动关系，可以使用不带箭头的关联实线。在参与者和用例之间的信息流不是由通讯关联来表示的，该信息流是缺省存在的（用例本身描述的就是参与者和系统之间的对话），并且信息流向是双向的，它与通讯关联箭头所指的方向毫无关系。

### 1.2 用例的内容

用例图使我们对系统的功能有了一

个整体的认知，我们可以知道有哪些参与者会与系统发生交互，每一个参与者需要系统为它提供什么样的服务。用例描述的是参与者与系统之间的对话，但是这个对话的细节并没有在用例图中表述出来，针对每一个用例我们可以用事件流来描述这一对话的细节内容。如在ATM系统中的"提款"用例可以用事件流表述如下：提款-基本事件流

1. 用户插入信用卡
2. 输入密码
3. 输入提款金额
4. 提取现金
5. 退出系统，取回信用卡

但是这只描述了提款用例中最顺利的一种情况，作为一个实用的系统，我们还必须考虑可能发生的各种其他情况，如信用卡无效、输入密码错、用户帐号中的现金余额不够等，所有这些可能发生的各种情况（包括正常的和异常的）被称之为用例的场景(Scenario)，场景也被称作是用例的实例(Instance)。在用例的各种场景中，最常见的场景是用基本流(Basic Flow)来描述的，其他的场景则是用备选流(Alternative Flow)来描述。对于ATM系统中的"提款"用例，我们可以得到如下一些备选流：提款-备选事件流

备选流一：用户可以在基本流中的任何一步选择退出，转至基本流步骤5。

备选流二：在基本流步骤1中，用户插入无效信用卡，系统显示错误并退出信用卡，用例结束。

备选流三：在基本流步骤2中，用户输入错误密码，系统显示错误并提示用户重新输入密码，重新回到基本流步骤2；三次输入密码错误后，信用卡被系统没收，用例结束。...

通过基本流与备选流的组合，就可以将用例所有可能发生的各种场景全部描述清楚。我们在描述用例的事件流的时候，就是要尽可能地将所有可能的场景都描述出来，以保证需求的完备性。

### 1.3 用例方法的优点

用例方法完全是站在用户的角度上（从系统的外部）来描述系统的功能

的。在用例方法中，我们把被定义系统看作是一个黑箱，我们并不关心系统内部是如何完成它所提供的功能的。用例方法首先描述了被定义系统有哪些外部使用者（抽象成为Actor），这些使用者与被定义系统发生交互；针对每一参与者，用例方法又描述了系统为这些参与者提供了什么样的服务（抽象成为Use Case），或者说系统是如何被这些参与者使用的。所以从用例图中，我们可以得到对于被定义系统的一个总体印象。 100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)