

## 第9章

# 通过用例规范化需求

当您准备外出度假时，可能会在心里或用纸笔列个清单：是否已经打点好行李？行李是否太多？有没有退订报纸、邮件等服务？是否已找人来给植物浇水和喂养宠物鼠？一旦开始行程，您希望自己玩得开心，而且不用担心回家的时候家里一团糟。

这类似于软件开发项目。您需要在开始开发前组织好系统需要提供的功能清单，这样项目才能平稳运行，而且不会在交付系统时发生灾难(例如，不满足需求，客户或用户不满意)。

需求分析的艺术和科学(很明显两者皆有)是个范围很广的课题，简直可以用整本书来探讨它。有一种用来发现和列举需求的特定技术，称为用例建模(use case modeling)。严格来说，用例不是面向对象(OO)方法学的产物，用例适用于任何软件系统，无论该系统使用的是何种开发方法。然而，用例是在面向对象系统的背景下首次亮相于软件开发社区，并且在面向对象开发中得到了人们的广泛接受。

### 本章主要介绍如下主题：

- 我们必须如何预测用户在与系统交互时所扮演的所有不同的角色
- 在描述软件应用程序作为一个整体来提供服务时，必须设想其中每个服务的视角
- 如何准备用例，以作为记录所有需求的手段

本书还会介绍有关需求分析的大体背景知识，从而让您了解用例建模。

## 9.1 用例的概念

当要确定系统所需的功能时，必须找到以下问题的答案：

- 使用系统的对象是谁？
- 系统应该提供哪些有价值的功能？
- 当用户出于某种特定目标而与系统进行交互时，他们希望得到什么结果？

用例(use case)可以回答上述问题。每个用例都是一个简单的声明，体现为文字或图表形式，它描述了系统的特定目标或结果，以及期望得到该结果的对象。例如，学生选课系统(SRS)的目标是“让学生选修课程”或“让教授同意执教课程”。这样，我们就已

经表达了第一个用例！用例真的就是如此直截了当。事实上，我们需要直截了当的用例，这样系统的用户或发起者才能理解用例，稍后将对此做进一步讨论。

### 9.1.1 功能需求和技术需求

考虑系统全部用例的目的在于彻底地研究系统的功能需求，从而确保没有忽视某类特定用户或系统的潜在目标。以下内容将区分功能需求和技术需求。

功能需求(functional requirements)是指从使用系统的用户角度看，与操作或功能有关的系统行为。功能需求可分为以下两种：

- 面向目标的功能需求(goal-oriented functional requirements): 这类需求描述系统目标，而不考虑特定的用户需求，例如“系统必须能提供自定义报表”。在指定面向目标的需求时，应该避免讨论实现的细节，例如“系统必须使用 CrystalReports 生成自定义报表”。在准备用例时强调的就是面向目标的功能需求。
- 体验需求(look and feel requirements): 这类需求比较明确，主要取决于用户期望的系统外观(如图形用户接口(GUI)的外观)以及用户从自己的角度出发所期望的系统行为。例如，可能会有这样的需求：用户在主界面上单击某个按钮，系统显示确认消息……一个良好的习惯是编写操作概念(concept of operations)文档，将其作为“纸上原型”，这样能在开始建模前模拟并和用户一起讨论将要建立的系统。



#### 注意

第 16 章将会介绍 SRS 应用程序的操作概念的示例。

技术需求(technical requirements)主要涉及系统如何在内部构建，以满足功能需求。例如，“系统必须和 TCP/IP 协议兼容……”您可以将这些需求看作是对技术的约束，而功能需求则是对要处理问题的声明。类似这样的技术需求在用例分析阶段没有用。

虽然系统用户可能精通技术，但最好以易于理解的方式表达功能需求，力求让不了解计算机内部运行原理的人们也能理解。这样能够确保技术需求不会混入功能需求声明中，这是很多缺乏经验的软件开发人员常犯的错误。如果让技术需求影响功能需求，则在开发生命周期中，技术需求将会过早地限制解决方案。

### 9.1.2 让用户参与

用户是最了解系统功能的最终专家，因此绝对有必要让用户参与用例的定义阶段。对于一些商业软件，如果没有定义具体的用户，则可以利用具有类似经验的“用户代理”来确定用户真正的需求。理想情况下，用户或用户代理将会自己编写部分的或全部的用例。至少您应该询问这些人员，站在他们的立场上编写需求，并且请他们确认用例是否准确。

用例是软件开发项目生命周期的第一步。用例经证明是编写测试脚本的基础，能够确保系统测试和用户可接受性测试中的所有功能线程均被执行。用例也是编写需求追踪

表(requirements traceability matrix)的基础,它是一个最终清单,当系统交付时,用户能够根据该清单验证每个初始需求是否被满足。当然,需求追踪表必须考虑系统的所有需求——无论是功能需求还是技术需求——而用例只是其中的一个子集。图 9-1 显示了在 Excel 工作簿中创建的需求跟踪表。

需求号	需求声明	发起者	测试说明	测试用例号	验证	修改
SRS1.1	遵循 W3C 和相关的标准	系统管理员	开发人员单元测试	1.1.2	已验证	无
SRS1.2	能够连接 ADO.NET 数据库	系主任	系统集成	1.2.0	未决	无

图 9-1 需求追踪表的示例

回答本章开始时提出的第一个问题“使用系统的对象是谁?”在用例术语中,这称为确定参与者(identifying actors)

## 9.2 参与者

参与者(actor)代表了系统建好后与之交互的对象,参与者的需求是用例的基础。参与者通常分为两大类:

- 人类用户
- 其他计算机系统

当参与者与用户交互时,他们通常想要达到某种结果,但也可以是向系统提供/贡献信息,或从系统接受/使用信息。

所谓提供信息,是指参与者输入信息,并由系统存储数据。例如,系主任安排一门新课程或学生选报课程计划。这不包括用户必须输入相对细节的信息以提供查询功能,例如,可通过输入学号来查询成绩。

所谓使用信息,是指参与者使用系统来获得信息。例如,教师用户打印选修其课程的学生名单,或学生在线查看其课程时间表。

### 9.2.1 确定参与者并定义其角色

必须为系统的每种角色创建一个参与者,这些角色代表了和系统有关的用户。要确定这些角色,首先需要查阅叙述性的需求说明书(narrative requirements specification),它是功能需求的声明,如 SRS 说明书。在 SRS 说明书中,唯一被明确提出的用户类别是学生用户,所以可将学生看作 SRS 的参与者类型。

如果不只是考虑说明书,不难发现使用 SRS 的其他用户类别:

- 教师可能需要了解选修他们课程的学生人数,或者可能使用系统来提交最终的分数,这些分数会反映在学生的成绩单上。
- 系主任可能想要知道哪些课程受欢迎,哪些课程比较冷清,是否需要取消一些选

修人数太少的课程。

- 选课办公室的工作人员可能想要使用 SRS 来证实某位学生是否修够了本学期的学分。
- 毕业生可能使用 SRS 打印他们的成绩单复印件。
- 新生——即打算申请入学但还没有完成申请的学生，他们可能想要浏览下学期将会提供的课程，从而确定这所大学是否能满足他们的需求。

同样，因为其他计算机系统也能成为参与者，所以可能也需要在 SRS 和其他已有的自动化系统之间建立接口，例如：

- 收费系统，这样能根据学生选修的课程来进行计费 and 付款
- 教室安排系统，保证不同容量的教室按照学生人数进行合理分配
- 入学系统，新生入学并符合申请课程的条件时，SRS 将得到通知

当然，必须尽早地确定系统的范围，避免需求膨胀或任务增多。如果很早就把所有可能的参与者都考虑进来，则对系统发起者来说成本会变得很高。例如，让学生使用 SRS 来预览学校开设的课程是否有意义？或者是否存在满足该需求的其他系统(例如，在线课程目录系统等)？通过与所有用户群的深入沟通，就能够正确地界定系统范围，而且还能剔除一些假定的参与者。

在本例中，将假设 SRS 系统的发起者在建立系统时不需要考虑毕业生或新生，也就是说，不把毕业生或新生看作是参与者。关键点在于，是发起者做决定，而不是程序员！软件工程师的责任之一是确定需求，也就是对系统提出对用户有利的功能改进意见。然而，只有发起者才真正有权利做定论。



#### 提示

很多软件工程师都会碰到麻烦，因为他们认为自己比客户更清楚其需求。您可能确实有不错的想法，但要牢记，只能把这种想法当作建议。如果不能让发起者/用户心悦诚服，那就准备好被他们回绝。

注意，同一个用户可能在不同的场合下以不同的角色和系统进行交互。例如，一位教授可能是系主任，他要决定是否取消某门课程；同时他也是一位教师，需要查询选修其课程的学生人数。

### 9.2.2 为系统和参与者绘制用例图

在确定系统的参与者后，可能需要绘制用例图。统一建模语言(Unified Modeling Language, UML)的符号如下：无论是用户或是计算机系统，都用小人代表所有的参与者，矩形表示系统，直线表示参与者和系统相连(如图 9-2 所示)。

这幅图显得相当简单，但它可能正是为 SRS 这样的项目所绘制的正确图纸。

这里将使用修改后的 UML 符号，如下所示：

- 使用矩形不仅表示核心系统，也可以表示作为外部的所有参与者，从而不用小人

表示后者。

- 使用箭头表明信息流的方向(无论参与者提供或使用信息), 这样更易于说明。例如, 在修改后的用例图中, 我们表示学生既提供信息也使用信息, 而选课员只能使用信息。

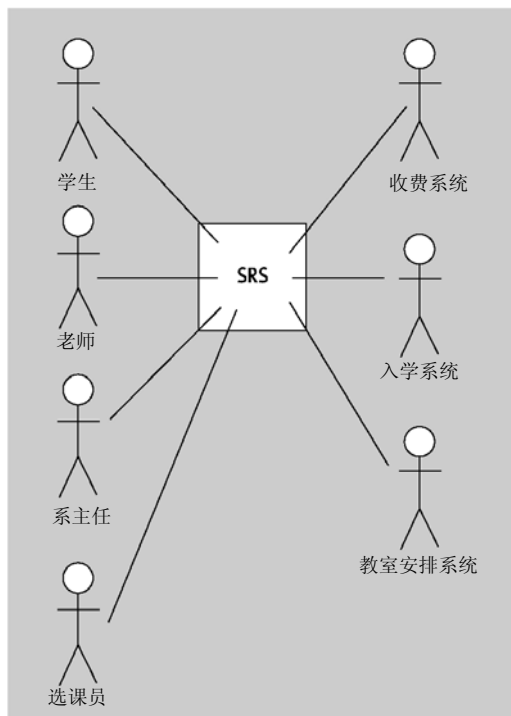


图 9-2 UML 用例图

### 注意

选课员确实提供了信息, 但是并不直接提供给 SRS。选课员向入学系统提供关于哪些学生在学校中注册的信息, 入学系统然后将这些信息传入 SRS。因此, 入学系统作为参与者向 SRS 提供信息, 但是从 SRS 的角度来看, 选课员只是一个使用者。

通过少量修改这些符号(如图 9-3 所示), UML 图就可以提供更多的信息。

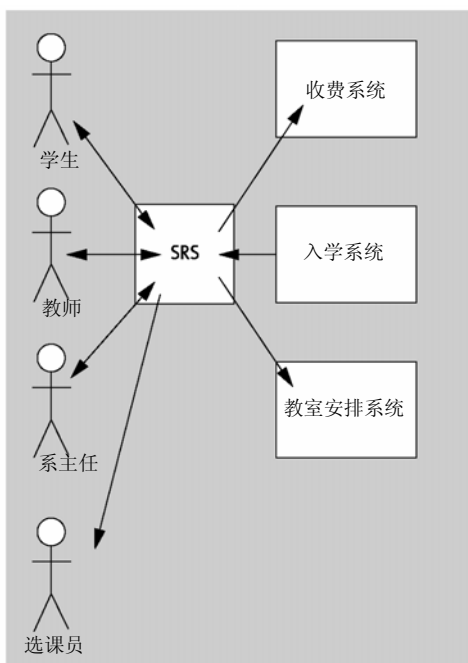


图 9-3 用例图符号的自定义版本

当然，如果您确定要略微修改像 UML 这样的被人们广泛接受的符号标准，则需要遵循以下步骤：

- (1) 和软件开发人员达成一致，保证团队使用相同的语言。
- (2) 将自定义的符号(包括整个符号)记录成文档，并和客户/用户沟通，从而让他们理解您的“语言”。
- (3) 确保这样的文档能集成到项目的完整文档集中，这样其他查看文档的人员能立即理解符号的含义。

如果只是凭直觉进行改进，那么它们可能只是一些自言自语的符号。当然，如第 8 章所述，您需要考虑 CASE 工具是否支持这样的修改。

本书的第 II 部分经常会提醒您，通过调整或扩展流程、符号或工具来满足公司或项目的需求，这样的做法是完全可以接受的，因为没有什么方法学组件是“神圣不可侵犯的”。

## 9.3 指定用例

在介绍完 SRS 的参与者后，接下来将会列举这些参与者使用系统的方式，即用例本身。

用例表示逻辑线程(thread)，或一系列因果事件，以参与者首次与系统交互为起点，以参与者使用系统而达到自己的目的为终点。注意，参与者始终会启动一个用例，系统自身启动的操作并不能保证用例的开发(如本章前面所述，虽然操作可能是操作需求或技



术需求)。

用例强调系统的功能，即功能需求，而不用考虑其内部实现方式，这类似于方法签名。事实上，可以将用例看作是系统的“行为签名”。

学生参与者的一些高级用例如下：

- 注册课程
- 取消课程
- 选择导师
- 建立学习计划
- 查看课程时间表
- 请求成绩单
- 确定学生的毕业资格

教师参与者的一些高级用例如下：

- 确定学生的课程
- 查看课程时间表
- 请求给定课程的学生名单
- 请求给定学生的成绩单
- 维护课程信息(例如，修改课程说明，更换任课教师等)
- 提交给定课程的学期成绩

查看以上用例，有时不同的参与者会共享相同的用例。例如，学生和教师都需要检查课程时间表。记住，用例由参与者发起，所以 SRS 需求说明书没有列举其他功能，例如，将“通过电子邮件通知学生”作为用例。

可以把用例分解为若干步骤，每个步骤表示更详细的用例，例如，注册课程可分解为以下步骤：

- (1) 验证学生是否符合选课的先决条件。
- (2) 检查学生的学习计划，确保需要该课程。
- (3) 检查课程是否有名额。
- (4) 将学生放入候选列表(可选)。

用例之间可能存在父-子关系，多个更通用的用例可能会共享某个更详细的用例。例如，“请求学生名单”和“提交最终成绩”这样的通用用例都会包含“验证该教授是否为本课程教师”的用例。

遗憾的是，和需求分析一样，没有好的方法能够保证已经确定了所有重要的用例或参与者，以及是否足够深入地细分子用例。用例的开发过程是迭代的，当后面的迭代不再发生新变化时，分析工作就可能已经完成！通过全面询问和回访用户、观察(跟踪)用户的日常任务和小组定期回顾所有用例，才能保证没有遗漏重要的用例。

## 9.4 匹配参与者和用例

接下来一个重要的步骤是将参与者和用例进行匹配。参与者和用例之间的关系潜在地是多对多的关系，也就是同一个参与者可能会启动不同的用例，或者单个用例可能和很多不同的参与者相关。通过参与者和用例之间的交叉引用，可以保证以下内容：

- 在最终分析中，不需要确定对系统毫无用处的参与者。
- 相反，不需要指定所有人都不关注的用例。
- 最重要的是，它能确保每个参与者关注于系统的某个方面，从而能在检查用例图时保证模型没有忽略重要的参与者。

对于每个用例-参与者的组合，有必要判断参与者是使用信息还是提供信息。另一种判断方法是：参与者需要有对系统信息资源的写访问权限(提供)还是只读访问权限(使用)。

如果对参与者和用例的数量没有限制，可用一个简单的表格来归纳上面的内容，如表 9-1 所示。

表 9-1 简单的参与者/用例交叉引用表

参与者	学生	教师	收费系统	其他
用例				
注册课程	提供信息	N/A	N/A	
提交最终成绩	使用信息	提供信息	N/A	
请求成绩单	使用信息	使用信息	N/A	
确定学生的课程计划	使用信息	使用信息	使用信息	
其他				

## 9.5 是否绘制用例图

用例的概念相当直接，因此简单的叙述文字对表达用例来说通常已够用了。然而，UML 提供了一种正式的方式来绘制用例以及用例和参与者之间的交互。前面提到过：

- 参与者用小人来表示
- 核心系统以及系统外部的所有参与者都使用矩形表示。

现在将加入以下内容：

- 用例使用下面带有简要说明文字的椭圆来表示。
- 椭圆外面的方框表示系统边界。

图 9-4 显示了一个样本 UML 用例图，该图描绘了 3 个参与者——学生、教师和选课员——各自参与“查询成绩”用例的情况。



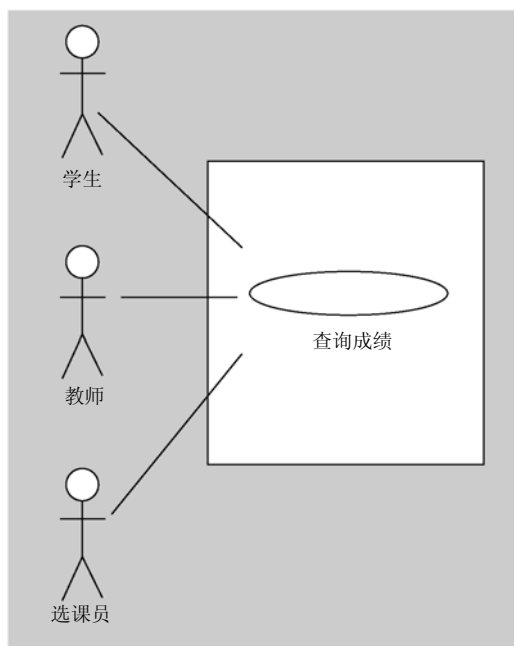


图 9-4 样本 UML 用例图

当要确定是用图来表示用例还是仅仅用文字叙述用例时，可首先考虑创建用例的基本原理：先全局考虑软件开发小组对系统需求的理解，然后与用户/发起者沟通并取得一致。用例图是否有助于这个过程完全取决于您、您的项目小组和您的用户/发起者。如果他们觉得有用，就用它；反之则编写用例文档，例如，本书前言部分最后的 SRS 需求说明书。

当您用文档记录系统的参与者和用例后，不管是文本形式还是搭配用例图，该描述都是定义要解决问题的核心文档的一部分。在第 10 章中将会介绍如何用这样的文档来确定创建和初始化系统所需的类。

UML 清楚地说明了和用例建模相关的一些约定：关于用例图的更多细节内容(包括绘图的高级技术)，请参阅第 17 章中推荐的阅读列表。

## 9.6 本章小结

在本章中，您学习了以下内容：

- 用例分析是一种精确、完整地指定系统功能需求的简单而强大的技术。
- 用例基于面向系统目标的功能需求。
- 用例用来描述：
  - 将要建立的系统的预期行为和功能
  - 使用这些服务的外部用户或系统(称为参与者)
  - 上述两者之间的交互

- 不属于用例的元素(如技术需求)

## 9.7 练习

- (1) 对于第2章练习(3)中定义的需求, 确定有哪些参与者。
- (2) 根据在第2章练习(3)中定义的需求, 列出(a)说明书将明确调用的用例; (b)系统的潜在用户可能用到的其他用例。
- (3) 创建一个表格将练习(1)中确定的参与者映射到练习(2)中列出的用例, 说明特定参与者在某个用例中是信息的提供者还是使用者。