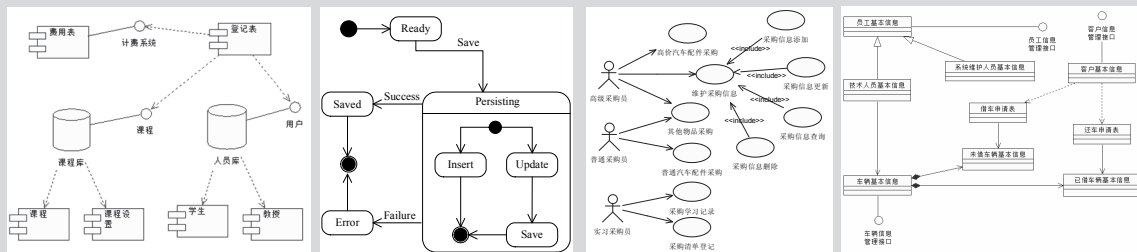


第 3 章

用例和用例图



用例是文本形式的情节描述，广泛应用于需求的发现和记录工作中。而用例图是指由参与者 (Actor)、用例 (Use Case) 以及它们之间的关系构成的用于描述系统功能的视图，是被称为参与者的外部用户所能观察到的系统功能的模型图。用例图是 UML 中较为重要和常用的一种图，由开发人员与用户经过多次商讨而共同完成，呈现了一些参与者和一些用例，以及它们之间的关系，主要用于对系统、子系统或类的功能行为进行建模。本章主要介绍用例图的概念、参与者和用例等一些基本概念及表示方法，以及用例图建模技术及应用技巧。

3.1 用例图的构成

用例图是一种将用例和软件工具相结合的图形表示方式，它由参与者发起，主要显示了一组用例、参与者以及它们之间的关系。

3.1.1 什么是用例图

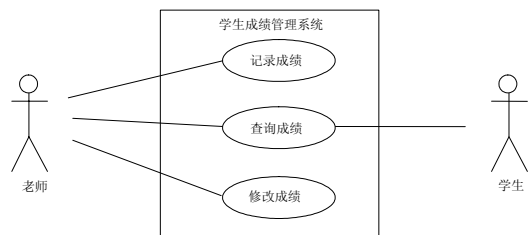
用例图是由软件需求分析到最终实现的第一步，它从用户角度来描述系统功能，描述系统的参与者与系统用例之间的关系。

用例图通常在进行需求分析时使用，由开发人员与用户经过多次商讨而共同完成，这些图以每一个参与系统开发的人员都可以理解的方式列举系统的业务需求。

用例图使用系统与一个或多个参与者之间的一系列消息来描述系统中的交互，它将系统功能划分为对参与者（系统的理想用户）有用的需求，其交互部分被称作用例。除此之外，用例图仅仅是从外部观察系统功能，也就是从参与者使用系统的角度描述系统中的信息，并不描述这些功能在系统内部的实现过程。

用例图不仅包含系统、参与者和用例 3 个元素，而且还包含表示这些元素之间存在的泛化关系、关联关系和依赖关系等各种关系。

用例图是描述参与者与系统的关系，因此用例图整体上分为 3 部分：参与者、系统和关系，通过关系将参与者与系统联系起来。下图描述了一个学生成绩管理系统的用例图，它是一个实际系统简化后的示例。



人形表示参与者；矩形为系统；椭圆形是用例；线条为关系，连接用例和参与者。在下面的小节中

将对上述元素作详细介绍。

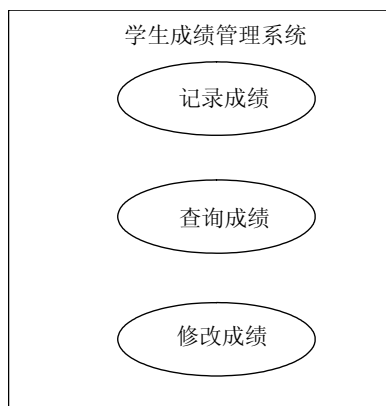
3.1.2 系统

系统是用例图的一个重要组成部分，用于执行特定功能。它不单指一个软件系统，而是为用户执行某类功能的一个或多个软件构件。如图书馆管理系统、学生选课系统、信息发布系统等都属于系统。

系统的边界用来说明用例图应用的范围。例如，系统拥有一定应用范围，例如一台自动售货机，提供售货、供货、提取销售款等功能，这些功能在自动售货机内的区域起作用，自动售货机外的情况将不考虑。

准确定义系统的边界并不总是很容易的，因为有些情况下，严格地划分哪些任务是由系统完成，而哪些是由人工或其他系统完成是很困难的。另外，系统最初的规模应有多大也应该考虑。一般的做法是，先识别出系统的基本功能，然后以此为基础定义一个稳定的、精确定义的系统架构，以后再不断地扩充系统功能，逐步完善系统。这样做可以避免由于系统太大，需求分析不易明确，从而导致浪费大量的开发时间。

系统在用例图中用一个长方框表示，系统的名称被写在方框上面或方框内。方框内包含了该系统用符号标识的用例，如下图所示。



3.1.3 参与者

参与者是系统外的一个实体，它代表了与系统交互的用户、设备或另一个系统。

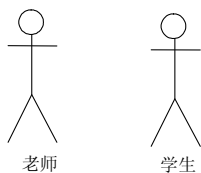
参与者是系统服务的对象，通过向系统输入信息或系统为参与者提供信息来进行交互，以实现系统功能。在确定系统的用例时，首要问题就是识别参与者。

1. 参与者的概念

参与者用于表示使用系统的对象。参与者可以是一个人、一个计算机系统、另一个子系统或另外一种对象。例如，计算机网络系统的参与者可以包括操作员、系统管理员、数据库管理员和普通用户，也可以有非人类参与者，如网络打印机。参与者的特征是其作为外部用户与系统发生交互。在系统的实际运作中，一个实际用户可能对应系统的多个参与者。同样，不同的多个用户也可以只对应于一个参与者，从而代表同一个参与者的不同实例。

每个参与者定义了一个角色集合，当系统用户与系统相互作用时会采用它们。参与者的一个集合完整描述了外部用户与系统通信的所有途径。当系统被实现时，参与者也被物理对象实现。物理对象如果可以满足多个参与者的角色，那么它就可以实现多个参与者。例如，一个人可以既是商店售货员又是顾客。这些参与者不是本质上相关的，但是它们可以由一个人来实现。当系统的设计被实施时，系统内的多个参与者被设计成类实现。

在用例图中，参与者由固定的图形表示，并在参与者下面列出参与者的角色名。当为用例图中参与者命名时，给作为系统用户的参与者提供一个最能描述其功能的合适名称是非常重要的。当为参与者命名时要避免为代表人的参与者起一个实际的人名，而应该以其使用系统时的角色为参与者命名。例如下图表示的参与者，老师表示所有以老师身份使用系统的人，而并不单指某个人。



参与者与系统的交互作用量化为用例，用例是设计系统和它的参与者连接的功能块，用来完成对参与者有意义的事情。一个用例可以被一个或多个参与者使用，同样，一个参与者也可以与一个或多个用例交互。最终，参与者由用例和参与者在不同用例中所担任的角色决定。没有参加任何用例的参与者是无意义的。

用例模型刻画了一个实体（如系统、子系统或类）与外部实体相互作用时产生的行为的特征。外部实体是实体的参与者。对于一个系统，参与者既可以由人类用户实现，也可以由其他系统实现。对于一个子系统或类，外部元素可以是整个系统的参与者，或者参与者可以是系统内的其他元素，如其他子系统或类。

在建模初期，参与者和用例交互，但是随着项目的进展，用例被类和组件实现，这时参与者也发生了变化。参与者不再是用户扮演的角色，而变成了用户接口。例如，系统分析阶段用例图中，图书管理员与借出书目用例交互，以借出某本图书。在设计阶段，该参与者就变成了两个元素，即图书管理员这个角色和图书管理员所使用的接口，用例在这时就变成了许多对象，负责处理与用户接口以及系统的其他部分交互。

2. 识别参与者

一个系统在建模之前虽然能确定一些用户和参与者，但并不能全面地不遗漏地将参与者找出，这将导致建模不完善、开发不完善，开发过程中的修改又将导致开发效率降低，漏洞产生。

全面识别参与者才能使建模很好地进行下去。为了能找出所有参与者，可以借助以下几个问题。

- 系统的主要客户是谁？
- 谁需要借助系统完成日常工作？
- 谁来安装、维护和管理系统，保证系统正常运行？
- 系统控制的硬件设备有哪些？
- 系统需要与哪些其他系统进行交互？
- 在预定的时刻，是否有事件自动发生？
- 系统是否需要定期产生事件或结果？
- 系统如何获取信息？

在寻找系统用户时,建模人员不应把目光只停留在使用计算机的人员身上,而应注意直接或间接地与系统交互或从系统中获取信息的任何人和任何事。在完成参与者的识别后,建模人员就可以从参与者的角度考虑参与者需要系统完成什么功能,从而建立参与者所需要的用例。

一个用例通常要与多个参与者发生交互。其中,不同的参与者所充当的角色不同;有些参与者接收用例所提供的数据,有些参与者则为用例提供某种服务,而另一些参与者要完成系统的管理。这就需要参与者分类,以保证把系统中所有用例都表示出来。

参与者通常可以被分为主要参与者与次要参与者两类。其中,主要参与者是使用系统较频繁、业务量较大的用户,系统建模人员在识别用例时应该首先识别主要参与者;次要参与者用来给用例提供某些服务。次要参与者与用例进行交互的主要目的是给其他参与者提供所需要的服务,也就是说,次要参与者要使用系统的次要功能。次要功能是指完成系统维护的一般功能。区分主要参与者与次要参与者不应该以参与者使用系统时的权限为依据,一般情况下,应该以使用系统时的业务量为依据。例如,在图书管理系统中,将参与者以主要与次要区分,可以将参与者分成图书管理员和系统管理员。其中,主要参与者负责图书的日常借阅任务,而次要管理者则完成对系统的维护。

除了对参与者进行主次区分外,还可以存在许多其他分类方法。例如,当参与者使用系统时,它们可能会承担着不同的“职责”,建模人员可以利用这些职责来定义参与者与系统间的交互,以及参与者在各种交互中所充当的角色。参与者在系统中的角色主要包括:

- 系统的启动者。
- 系统的服务者。
- 系统服务的接收者。

参与者在系统中所扮演的第一种角色是系统的启动者。启动者是系统的外部实体,它们是为了完成某项事务而启动系统的。一个启动者可以请求某种服务或者触发一个事件。例如,一个使用自动

提款机提款的用户就是该系统的一个启动者。

参与者所能承担的第二种角色就是系统服务者,服务者也是系统的外部实体,它们响应系统的请求,为系统提供某种服务。例如,在自动提款机的提款事件中,自动提款机系统需要银行的内部系统提供用户的存款信息。这个银行内部系统就是一个为系统提供服务的参与者。

系统服务接收者的主要职责是接收来自系统的信息。例如,使用自动提款机的用户就是自动提款机系统服务的接收者。从这个示例可以看出,一个人可以在系统中扮演不同的参与者。

参与者的分类方式很多,最终目的就是全面不遗漏地找出参与者。在对参与者建模的过程中,开发人员必须牢记以下几点。

- 参与者对于系统而言总是外部的,因此它们可以处于人的控制之外。
- 参与者可以直接或间接地同系统交互,或使用系统提供的服务以完成某件事务。
- 参与者表示人和事物与系统发生交互时所扮演的角色,而不是特定的人或特定的事物。
- 一个人或事物在与系统发生交互时,可以同时或不同时扮演多个角色。
- 每一个参与者需要具有一个与业务一样的名字,在建模中不推荐使用类似于“NewActor”或“新参与者”的名字。
- 每一个参与者必须有简短的描述,从业务角度描述参与者是什么。
- 和类一样,参与者可以具有表示参与者的属性和可以接受的事件,但使用得不频繁。
- 多个参与者之间可以具有与类之间相同的关系。

在完成参与者的识别工作后,建模人员就可以从参与者的角度出发,考虑参与者需要系统完成什么样的功能,从而建立参与者所需要的用例。

3.1.4 用例

用例可以是一组连续的操作,也可以是一个特定功能的模块。系统由一个或多个用例构成,参与

者与系统的关系主要表现在参与者与系统用例的关系。用例是一个叙述型的文档，用来描述参与者使用系统完成的事件。

1. 用例的概念

用例是用户期望系统具备的功能，它定义了系统的行为特征。用例的目标是要定义系统（包括一个子系统或整个系统）的一个行为，但并不显示系统的内部结构。每个用例说明一个系统提供给它的使用者的一种服务，即一种对外部可见的使用系统的特定方式。它以用户的观点描述用户和系统间交互的完整顺序，以及由系统执行的响应。这里的交互只包括系统与参与者之间的通信，而其内部行为和实现是隐藏的。一个系统的全部用例分割和覆盖它的行为，每个用例代表一部分量化的、有深刻意义的和对用户可用的功能。

命名用例与命名参与者同样重要。用例名可以是带有数字、字母和除保留符号——冒号以外的任何标点符号的任意字符串。一般情况下，命名一个用例时要尽量使用动词加可以描述系统功能的名词。例如，提取货款、验证身份等用例，其侧重点是目标，而不是处理过程。

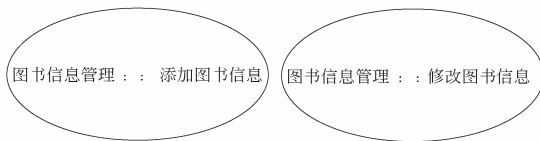
在 UML 中，用例用一个椭圆来表示，用例的名称可以写在椭圆的内部，也可以写在椭圆的外部，但通常情况下是将其名称写在椭圆内部，如下图所示。



需要注意，一定不要在一个用例图中使用两种命名方法，即将用例名写在椭圆之外和椭圆之内。因为这很容易会让模型的读者产生混淆。

一个系统完整的用例描述了该系统的所有行为，这可能导致用例图中的用例非常庞大。为了组织建模信息，UML 提供了包的概念，它的功能和目录相似。为了便于使用，可以把一些相关的用例放在一个包中。这样包就变成了包括相关功能的系统的子集。可以通过在用例前面加上包名和两个冒

号来确定该用例是属于哪个包的，如下图所示。



注意

用例只会指出系统应该做什么，即系统的需求，而不是确切地说明系统不必做什么，即系统的非功能需求。

2. 识别用例

系统分析者必须分析系统的参与者和用例，它们分别描述了“谁来做”和“做什么”这两个问题。

识别用例最好的方法就是从分析系统的参与者开始，对于已经识别的参与者，通过考虑每个参与者是如何使用系统的，以及系统对事件的响应来识别用例。使用这种策略的过程可能会发现新的参与者，这对完善整个系统的模型是有很大帮助的。用例模型的建立是一个迭代过程。

在识别用例的过程中，通过询问下列问题就可以发现用例。

- 参与者需要从系统中获取哪种功能，即参与者要系统“做什么”？
- 参与者是否需要读取、产生、删除、修改或存储系统中的某种信息？
- 系统的状态改变时，是否通知参与者？
- 是否存在影响系统的外部事件？
- 系统需要什么样的输入/输出信息？

在用例识别中需要注意以下问题。

- 用例图中每个用例都必须有一个唯一的名字以区别于其他用例。
- 每个用例的执行都独立于其他用例。
- 用例表示系统中所有对外部用户可见的行为。
- 用例不同于操作，用例可以在执行过程中持续接受或持续输出与参与者交互的信息。



用例的识别也可以通过查找事件的方式来确定,即找出参与者使用系统时的所有操作及获取信息,列为事件表,再根据事件表确定系统用例。

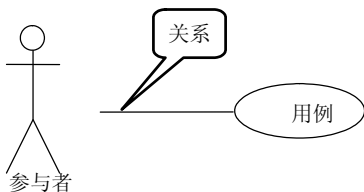
用例图有以下4种标准关系。

- 泛化关系 参与者间或用例间的关系,类似于继承关系,可以重载。
- 关联关系 参与者与用例间的关系。
- 包含关系 用例与用例的关系,将复杂的用例分解成小的步骤用例。
- 扩展关系 用例间的关系。

3.1.5 关系

这里讲的关系是参与者与用例间的关系,即关联关系。用例图就是描述系统和参与者关系的,而用例和参与者都是独立的事物,关系就是它们之间的关联或通信。这种通信是双向的,参与者肯定要参与某个或多个用例交互,用例也肯定会有参与者与之交互,否则参与者或用例将会成为多余。

使用一条实线连接参与者与用例,即可表明它们的关系,如下图所示表示了一个用例图中的关系。



这个简单的示例只显示了参与者与用例之间的一条通信关联。

不同的参与者可以访问相同的用例,一般来说它们和该用例的交互是不一样的。如果一样的话,那么参与者可能要重新定义。如果两种交互的目的也相同,说明它们的参与者是相同的,可以将它们合并。

用例描述系统满足需求的方式。当细化描述用例操作步骤时,就可以发现有些用例以几种不同的模式或特例在运行,而有些用例在整个执行期间会出现多重流程。如果将用例中重要的可选性操作流程从用例中分隔出来,以形成一个新的用例,这对整个系统的好处是显而易见的。

当分离可重复使用的用例后,用例之间就存在着某种特殊关系。包含和扩展是两个用例紧密相关时关联用例的两种方法。包含关系用于表示用例执行其功能时需要从其他用例引入功能。类似地,扩展关系则表示用例的功能可以通过其他用例的功能得到扩充。

除此之外,用例与用例之间也可以有继承关系,这种关系在用例图中称作泛化关系。在泛化关系中,子用例从父用例处继承行为和属性,还可以添加、覆盖或改变继承的行为,这对后期的开发很有用。

3.2 用例关系和描述

用例除了与其参与者发生关联外,还可以具有系统中的多个关系,这些关系包括包含关系、扩展关系和泛化关系,而应用这些关系的目的是从系统中抽取出公共行为和其变体。

3.2.1 泛化关系

泛化是一种表示UML中项目的继承关系的技术。泛化可以应用于参与者和用例中表示其子项从父项继承的功能,而且泛化还表示了父项的每个

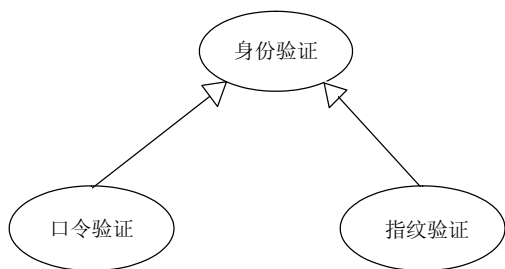
子项都有略微不同的功能或目的,以确保自己的唯一性。泛化可以用于用例,也可以用于参与者。

1. 泛化用例

相对于参与者而言,用例泛化更易理解。用例泛化是指一个用例(一般为子用例)和另一个用例(父用例)之间的关系,其中父用例描述了子用例与其他用例共享的特性,而这些用例是有着同一父用例的。

泛化将特化用例和一般用例联系起来。即子用例是父用例的特化，子用例除具有父用例的特性外，还可以有自己的另外特性。父用例可以被特化成一个或多个子用例，然后用这些子用例来代表父用例的更多明确的形式。

泛化的标记非常简单，它使用一条实线和三角箭头连接父用例和子用例，由子用例指向父用例，如下图所示。



因为父用例“身份验证”是抽象的，它并不提供具体的身份验证方法，所以一个具体的子用例必须提供具体的功能。

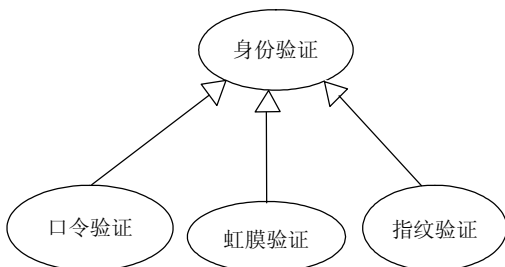
子用例“口令验证”提供的身份验证方法为：

- 从主数据库获得密码。
- 请求使用密码。
- 用户提供密码。
- 在用户登录时检查密码。

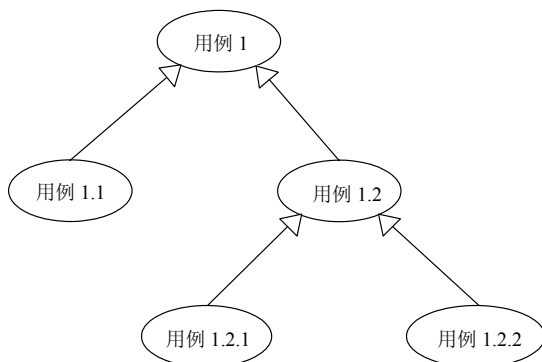
子用例“指纹验证”提供的身份验证方法为：

- 得到来自主数据库的指纹特征。
- 扫描用户的指纹特征。
- 将主数据库指纹特征和扫描特征比较。

深入研究本示例，会发现身份验证不只有两个子用例。下图演示了一个父用例的多个子用例。

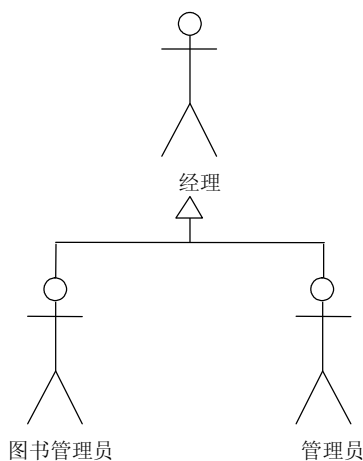


泛化甚至可以分层，父用例的子用例也可以有自己的子用例，如下图所示。

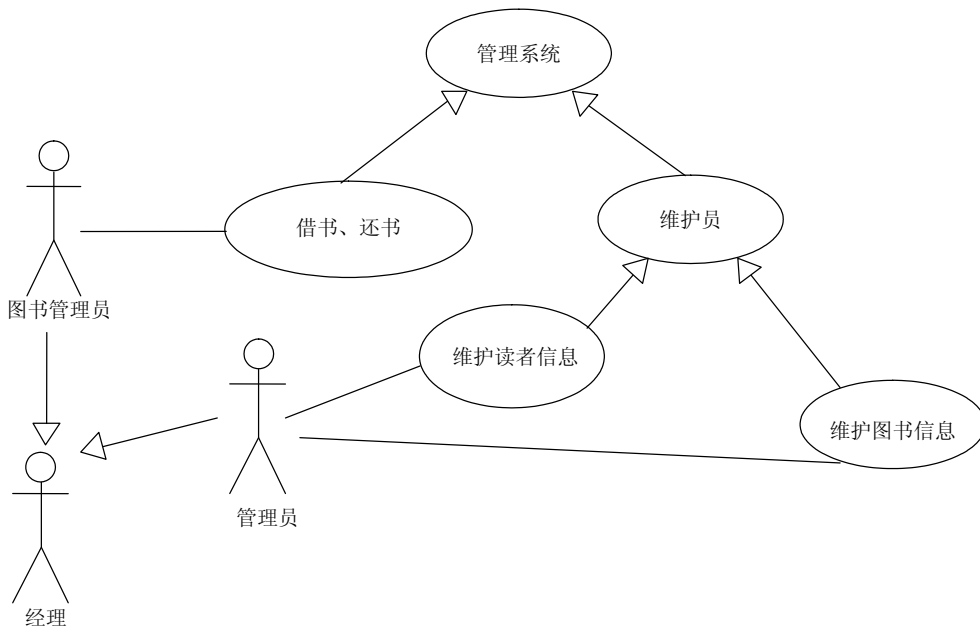


2. 泛化参与者

与用例一样，也可以对参与者进行泛化。泛化后的参与者在系统中扮演较为具体的角色。如下图所示，假设图书管理系统中，管理员分为对系统进行维护的管理员和完成借书、还书等日常操作的图书管理员。参与者“经理”描述了参与者“图书管理员”和“管理员”所扮演的一般角色。如果不考虑与系统交互时的职责，可以使用一般角色的参与者“经理”。如果强调管理员的职责，那么用例须使用精确的参与者，即子类“图书管理员”和“管理员”。



除此之外，还可以将泛化后的用例与泛化后的参与者相联系起来，如下图所示的泛化的用例与泛化的参与者相关联。



3.2.2 包含关系

在对系统进行分析时,通常会发现有些功能在不同的环境下都可以被使用。在编写代码时,希望编写可重用的构件,这些构件包括诸如可以从其他代码中调用或参考的类库、子过程以及函数。虽然每个用例的实例都是独立的,但是一个用例可以用其他的更简单的用例来描述。用例图中 UML 包含关系就支持这种做法。

包含关系指:一个用例可以简单地包含其他用例具有的行为,并把它所包含的用例行为作为自身行为的一部分。这种情况下,新用例不是初始用例的一个特殊例子,并且不能被初始用例所代替。包含关系把几个用例的公共步骤分离成一个单独的被包含用例。

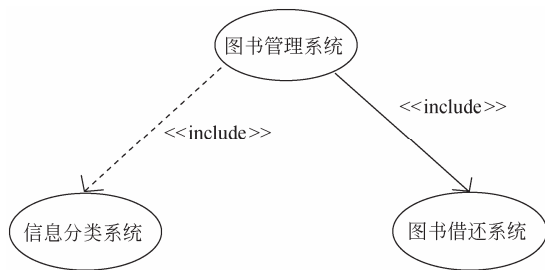
如果两个以上用例有大量一致的功能,则可以将这个功能分解到另一个用例中。其他用例可以和这个用例建立包含关系。

一个用例的功能太多时,可以用包含关系建模两个小用例。

被包含用例称作提供者用例,包含用例称作客户用例,提供者用例提供功能给客户使用。

在 UML 中,包含关系表示为虚线箭头加 <<include>> 字样,箭头指向被包含的用例。如下

图所示是图书管理系统中的包含关系。



为了更好地理解包含关系是如何起作用的,下面列出了“商品信息系统”和“建材信息系统”使用已经存在的被包含用例,如下图所示。



为了使用包含关系,用例必须遵循以下两个约束条件。

- ❑ 客户用例只依赖于提供者用例的返回结果,不必了解提供者用例的内部结构。
- ❑ 客户用例总会要求提供者用例执行,对提供者用例的调用是无条件的。

在为系统建模时,使用包含关系是十分明智

的。因为它有助于在将来实现系统时，确定哪里可以重用某些功能，在编写代码时就可实现代码的重用，从而从长远意义上缩短系统的开发周期。

3.2.3 扩展关系

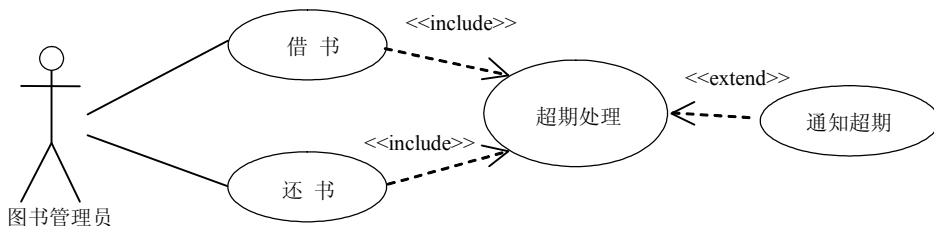
扩展关系是一种依赖关系，它指一个用例可以增强另一个用例的功能，是把新的行为插入到已有用例中的方法。

基础用例的扩展增加了原有的功能，此时是基础用例被作为用例使用，而不是扩展用例。

基础用例提供了一组扩展点，在这些新的扩展点中可以添加新的行为，而扩展用例提供了一组插入片段，这些片段能够被插入到基础用例的扩展点上。

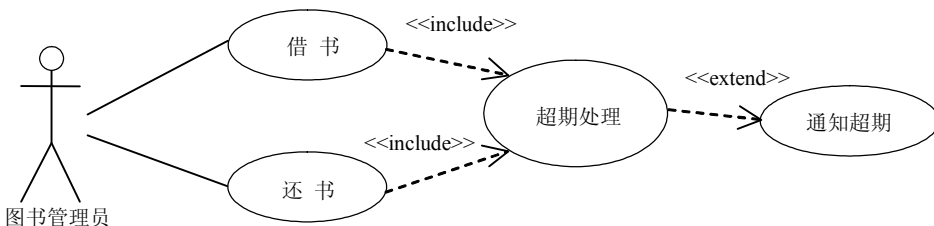
基础用例不必知道扩展用例的任何细节，它仅为其提供扩展点。

基础用例即使没有扩展用例也是完整的，这点



在本示例中，基础用例是超期处理，扩展用例是通知超期。如果借阅者按时归还图书，那么就不会执行通知超期用例。而当归还图书时超过了规定的时间，则超期处理用例就会调用通知超期用例提醒管理员对此进行处理。

正如上图中所表示的，通知超期用例指向超期



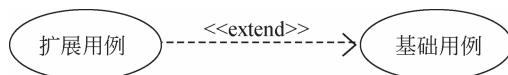
在理解了什么是扩展用例，以及使用它的原因后，那么如何知道图书管理员何时被提醒呢？毕竟这只在所借阅的图书超期时才被提醒，而且不是随时提醒的。本示例设定为当某学生所借阅的图书中

与包含关系有所不同。

一个用例可能有多个扩展点，每个扩展点也可以出现多次。一般情况下，基础用例的执行不会涉及扩展用例，只有特定的条件发生，扩展用例才被执行。

扩展关系为处理异常或构建灵活的系统框架提供了一种十分有效的方法。

在 UML 中，扩展关系表示为虚线箭头加 <<extend>> 字样，箭头指向被扩展的用例（即基础用例），箭头的尾部则处在扩展用例上，如下图所示是扩展关系标识符。

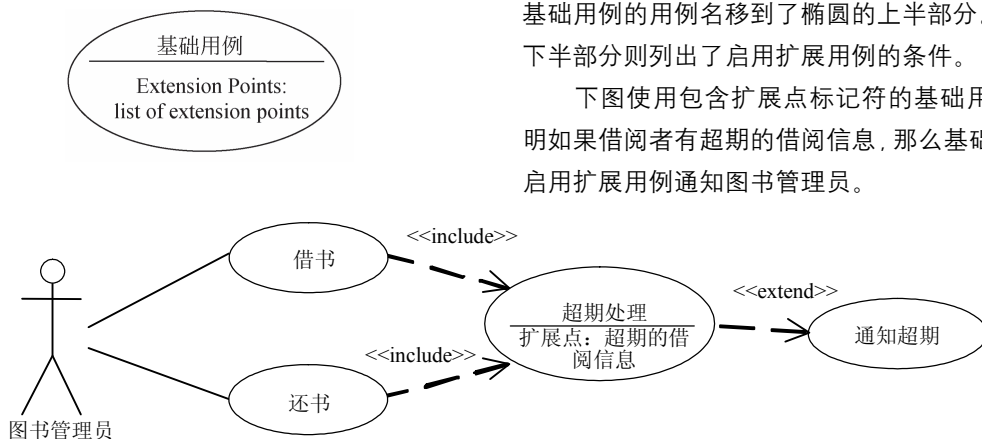


下面的示例将演示在图书管理系统中如何使用扩展关系：超期处理用例由通知超期用例进行扩展，如下图所示。

处理用例。这样绘制的原因是因为通知超期用例扩展了超期处理用例，即通知超期用例是添加到超期处理用例中的一项功能，而不是超期处理用例每次都调用通知超期用例。如果每次检查是否超期时都要提醒图书管理员，那么就要使用如下图所示的包含关系。

有超期借阅时，图书管理员才会被提醒。为此，UML 提供了扩展点来解决该问题。扩展点的定义为：基础用例中的一个或多个位置，在该位置会衡量某个条件以决定是否启用扩展用例。下图为一个

扩展点的标记符。



如上图所示，扩展点中有一个判断条件，以决定扩展用例是否会被使用，在包含关系中没有这样的条件。扩展点定义了启用扩展用例的条件，一旦该条件满足，则扩展用例将被使用。例如，当某学生的借阅信息中有超期的借阅信息时，则基础用例 ProcessOverTime 会使用 NotifyOverTime 用例，以通知图书管理员该学生有图书超期未还。当执行完扩展用例 NotifyOverTime 后，基础用例将继续执行。

扩展点的表示符号可以按照下面的格式添加到椭圆中，即：

```
<extension point>::=<name>
[:<explanation>]
```

其中，name 指扩展点的名称，因为一个基础用例可以有多个扩展用例。扩展点的名称描述了用例中的某个逻辑位置。因为用例描述的是功能和行为，所以该位置通常是对象在执行过程中某时间的状态。explanation 为对扩展点的解释，它为一个可选项。该项可以是任何形式的文本，只要把问题交代清楚即可。需要注意，在绘制扩展点时，并不是所有的 UML 建模工具都支持上述命名方法。

除在基础用例上使用扩展点控制什么时候进行扩展外，扩展用例自身也可以包含条件。扩展用例上的条件是作为约束使用的，在扩展点成立的时候，如果该约束表达式也得到了满足，则扩展用例才执行，否则不会执行。

如上图所示，一个水平线分隔了基础用例，而基础用例的用例名移到了椭圆的上半部分。椭圆的下半部分则列出了启用扩展用例的条件。

下图使用包含扩展点标记符的基础用例来表明如果借阅者有超期的借阅信息，那么基础用例则启用扩展用例通知图书管理员。

3.2.4 用例描述

用例图描述了参与者和系统特征之间的关系，但是它缺乏描述系统行为的细节。所以一般情况下，还会以书面文档的形式对用例进行描述，每个用例应具有一个用例描述。在 UML 中对用例的描述并没有硬性规定，但一般情况下用例描述应包括以下几个方面内容。

1. 名称

名称无疑应该表明用户的意图或用例的用途，如上面示例中的“借阅图书”“归还图书”。

2. 标识符[可选]

唯一标识一个用例，如“UC200601”。这样就可以在项目的其他元素（如类模型）中用它来引用这个用例。

3. 参与者[可选]

与此用例相关的参与者列表。尽管这则信息包含在用例本身当中，但在没有用例图时，它有助于增加对该用例的理解。

4. 状态[可选]

指示用例的状态，通常为以下几种之一：进行中、等待审查、通过审查或未通过审查。

5. 频率

参与者使用此用例的频率。

6. 前置条件

一个条件列表。前置条件描述了执行用例之前

系统必须满足的条件。这些条件必须在使用用例之前得到满足。前置条件在使用之前，已经由用例进行过测试。如果条件不满足，则用例不会被执行。

前置条件非常类似于编程中的调用函数或过程，函数或过程在开始部分对传递的参数进行检测。如果传递的参数无法通过合法检查，那么调用的请求将会被拒绝。同样这也适用于用例。例如，当学生借阅图书时，借出图书用例需要获取学生借书证信息，但如果学生使用了一个已经被注销的借书证，那么用例就不应该更新借阅关系；另外，如果学生归还了从系统中已经删除的一本图书，那么用例就不能让还书操作完成。

借阅图书用例的前置条件可以写成下面的形式。

前置条件：学生出示的借书证必须是合法的借书证。

7. 后置条件

后置条件将在用例成功完成以后得到满足，它提供了系统的部分描述。即在前置条件满足后，用例做了什么？以及用例结束时，系统处于什么状态？因为并不知道用例终止后处于什么状态，因此必须确保在用例结束时，系统处于一个稳定的状态。例如，当借阅图书成功后，用例应该提供该学生的所有借阅信息。

借阅图书用例的后置条件可以写成下面的形式。

后置条件：借书成功，则返回该学生借阅信息；借书失败，则返回失败的原因。

8. 假设[可选]

为了让一个用例正常运行，系统必须满足一定的条件，在没有满足这些条件之前，系统不会调用该用例。假设描述的是系统在使用用例之前必须满足的状态，这些条件并没有经过用例的检验，用例只是假设它们为真。例如，身份验证机制，后继的每个用例都假设用户是在通过身份验证以后访问用例的。应该在一定的时候检验这些假设，或者将它们添加到操作的基本流程或可选流程中。

下面是借阅图书用例的假设条件。

假设：图书管理员已经成功登录到系统。

9. 基本操作流程

参与者在用例中所遵循的主逻辑路径。因为它描述了当各项工作都正常进行时的用例的工作方式，所以通常称其为适当路径或主路径。操作流程描述了用户和执行用例之间交互的每一步。描述操作流程是一项将个别用例进行合适细化的任务。通过这种做法，常常可以发现自己原始的用例图遗漏了哪些内容。

借出图书用例的基本操作流程如下。

- (1) 管理员输入借书证信息。
- (2) 系统要确保借书证信息的有效性。
- (3) 检查是否有超期的借阅信息。
- (4) 管理员输入要借阅的图书信息。
- (5) 系统将学生的借阅信息添加到数据库中。
- (6) 系统显示该学生的所有借阅信息。

10. 可选操作流程

可选操作流程包括用例中很少使用的逻辑路径，那些在变更工作方式、出现异常或发生错误的情况下所遵循的路径。例如，借出图书用例的可选操作流程包括：输入的借书证信息不存在，该借书证已经被注销或有超期的借阅信息等异常情况下，系统采取的应急措施。

11. 修改历史记录[可选]

修改历史记录是关于用例的修改时间、修改原因和修改人的详细信息。下表是一个对用例“归还图书”的描述。

用例名称	归还图书
标识符	UC0002
用例描述	图书管理员收到要归还的图书，进行还书操作
参与者	图书管理员
状态	通过审查
前置条件	图书管理员登录进入系统
后置条件	在库图书数目增加
基本操作流程	(1) 系统管理员输入图书信息； (2) 系统检索与该图书相关的借阅者信息； (3) 系统检索该借阅者是否有超期的借阅信息； (4) 删除与该图书相关的借阅信息

续表

用例名称	归还图书
可选操作流程	该借阅者有超期的借阅信息,进行超期处理;输入的图书信息不存在,图书管理员进行确认
假设	图书管理员已经成功地登录到系统
修改历史记录	刘丽,定义基本操作流程,2006年10月20日 张鹏,定义可选操作流程,2006年10月22日

上表所示的格式和内容只是一个示例,开发人员

可以根据自己情况定义。但要记住,用例描述及它们所包含的信息,不仅是附属于用例图的额外信息。事实上,用例描述让用例变得完整,没有用例描述的用例没什么意义。

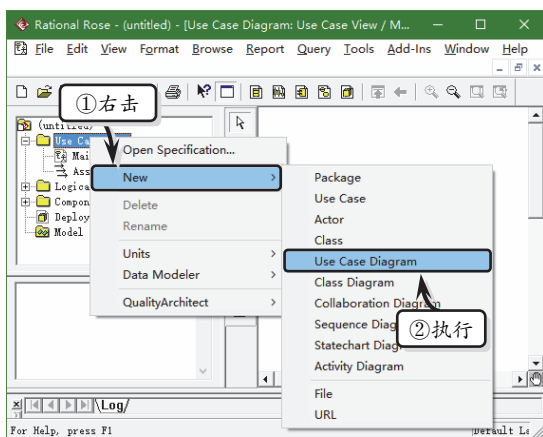
随着更多的用例细节被写到用例描述中,往往还会发现用例图中遗漏的某些功能。在模型的各个方面也会出现同样的问题:加入的细节越多,越可能必须回头更正以前所做的事。这是一个反复系统开发工作的内涵。进一步精炼系统模型是件好事,开发工作的每一次反复,都可以使系统模型更好、更准确。

3.3 绘制用例图

通过用例图可以有效地体现与理解客户的需求,在前面已经介绍了用例、用例图以及相关的一些概念。在本小节中将结合这些基础知识,介绍运用 Rose 软件创建用例图的方法。

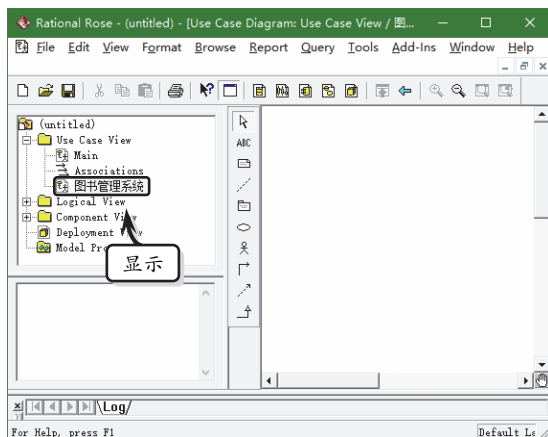
3.3.1 新建用例图

启动 Rose 软件,在“浏览器窗口”中的【Use Case View】上右击鼠标,执行【New】|【Use Case Diagram】命令。



此时,系统会自动在【Use Case View】栏下添加新创建的用例图,其默认名为“NewDiagram”。默认情况下,用例图的名称处于激活状态,可直接

输入新名称,重命名用例图名称。



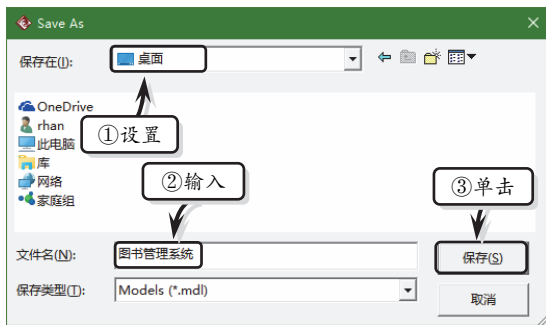
重命名用例图名称之后,双击该用例图模型,打开模型化窗体。【工具箱】包含了常用的 10 种工具,分别为选择工具 、文本 、注释 、注释锚 、包 、用例 、角色 、关系 、依赖 和泛化 。

提示

用户还可以通过执行【Tools】|【Create】命令,来选择相应的工具。

创建用例图模型后,执行【File】|【Save】命令,在弹出的【Save As】对话框中,设置保存位

置和名称，单击【保存】按钮，保存模型。

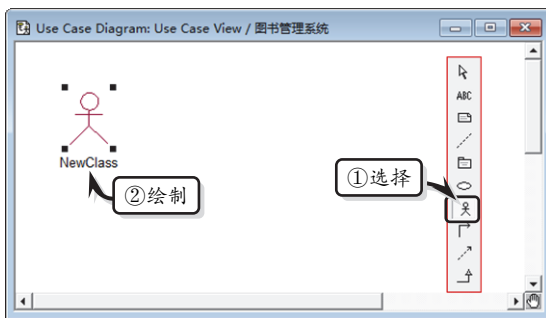


3.3.2 创建内容

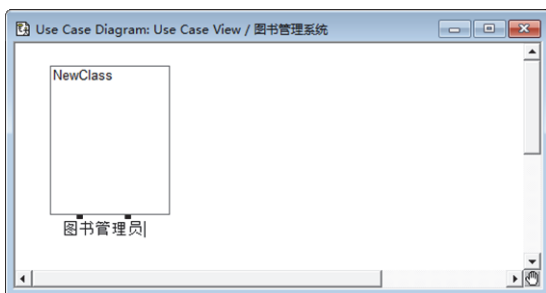
创建用例图模型后，便可以添加用例元素了。下面以“图书管理系统”为例，详细介绍添加用例元素的操作方法。

1. 添加参与者

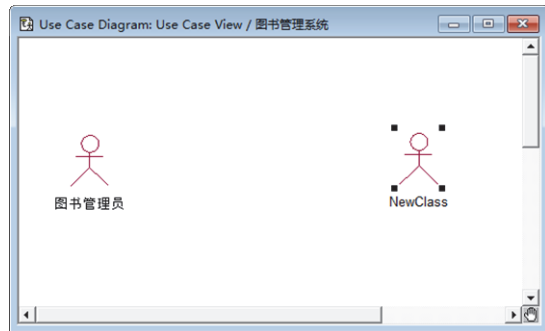
选择【工具箱】中的【Actor】工具，在“模型图窗口”中拖动鼠标绘制参与者图标。



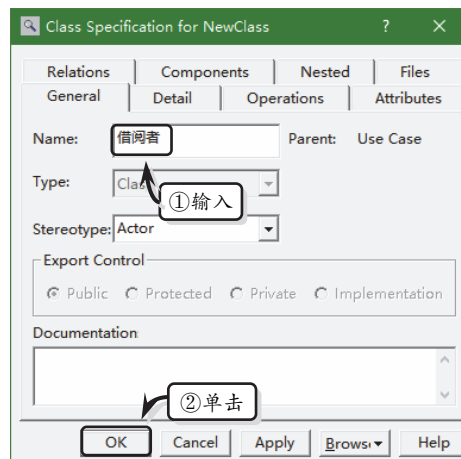
单击参与者图标名称，激活名称段，输入“图书管理员”文本，单击其他位置，完成重命名操作。



选择【工具箱】中的【Actor】工具，在“模型图窗口”中拖动鼠标绘制第 2 个参与者图标。



双击新建参与者图标，在弹出的【Class Specification for NewClass】对话框中，将【Name】选项更改为“借阅者”，并单击【OK】按钮。

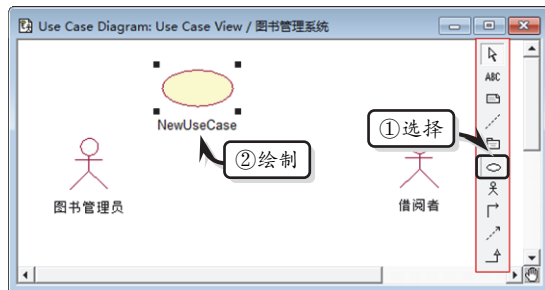


技巧

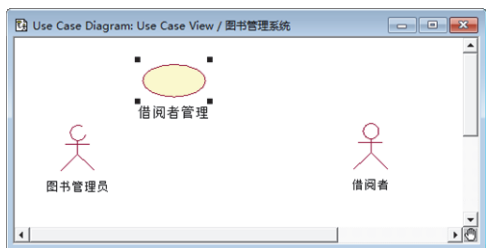
在“浏览器窗口”中右击【Use Case View】名称，执行【New】|【Actor】命令，即可创建参与者元素。

2. 添加用例

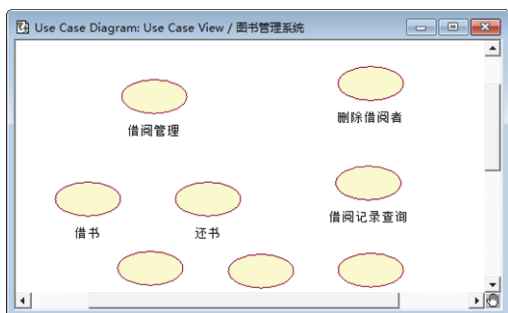
选择【工具箱】中的【Use Case】工具，在“模型图窗口”中拖动鼠标绘制用例图标。



单击用例图标名称，激活名称段，输入“借阅者管理”文本，单击其他位置，完成重命名操作。



使用同样的方法，创建其他用例，重命名用例并排列用例。



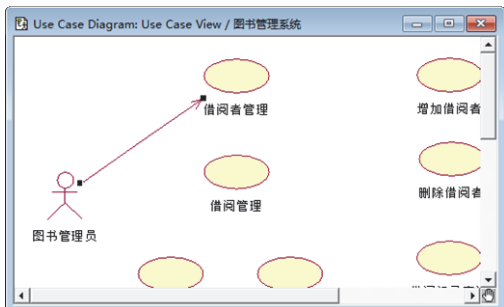
技巧

在“浏览器窗口”中右击【Use Case View】名称，执行【New】|【Use Case】命令，即可创建用例元素。

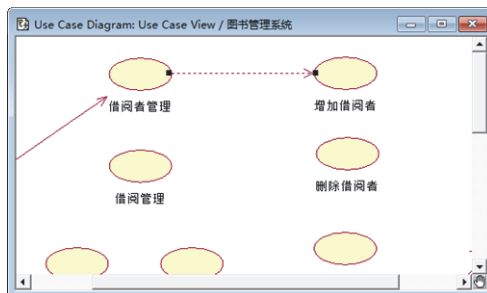
3. 添加关系

创建用例图的基础元素后，需要创建元素之间的关联。

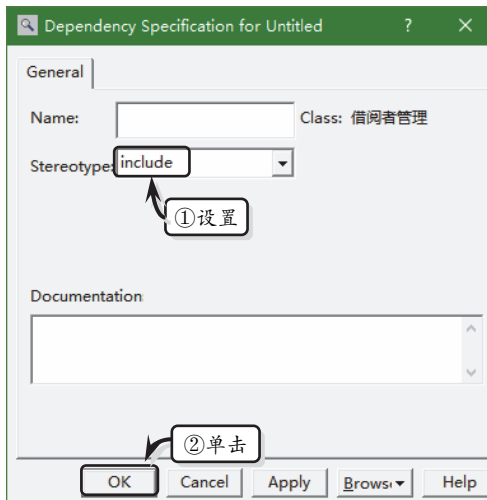
选择【工具箱】中的【Unidirectional Association】工具，将鼠标放在“图书管理员”上方，拖动鼠标至“借阅者管理”上方，松开鼠标即可。同样方法，添加其他关系。



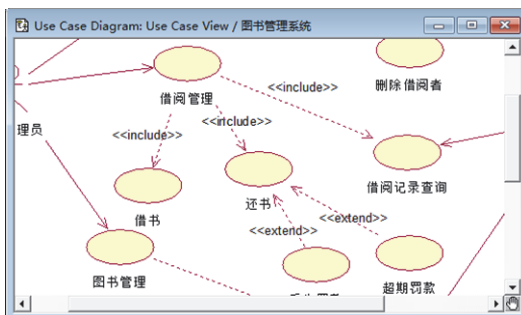
选择【工具箱】中的【Dependency or instantiates】工具，将鼠标放置在“借阅者管理”上方，拖动鼠标至“增加借阅者”上方，松开鼠标即可。



然后，双击 Dependency or instantiates 连接线，在弹出的【Dependency Specification for Untitled】对话框中，将【Stereotype】选项设置为“include”，并单击【OK】按钮。



使用同样方法，添加其他“include”和“extend”类型的依赖关系。



3.4 建模实例：创建 BBS 论坛用例图

论坛也叫网络论坛 (Bulletin Board System 或 Bulletin Board Service, BBS), 它们还可以称作电子公告板或公告板服务。论坛是 Internet 上的一种电子信息服务系统, 它提供一块公共的电子白板, 每个用户都可以在上面书写, 也可以发布信息或提出看法。本节以一个简单的论坛管理系统为例, 来说明用例图的创建过程。

3.4.1 确定系统信息

BBS 论坛中, 用户首先通过论坛登录网页 (如果是游客则需要注册) 进入论坛, 登录成功后可以通过发帖发布新的话题, 也可以对已经存在的话题进行回复, 还可以通过搜索来查看自己所关心的话题等。

在一个完整的论坛系统中可以实现多个功能, 如发帖、回帖、查看帖子以及注册登录功能。如下列出了比较常用且比较重要的论坛常用功能。

1. 普通用户注册成会员

几乎所有的网站都提供了用户注册成为会员的功能, 当然论坛系统也不例外。用户在系统注册页面可以填写自己的基本信息, 注册成功后系统会将信息保存到后台数据库中。另外, 注册成功后用户也可以查看和修改当前的内容。

2. 会员用户登录

论坛系统中提供了会员用户登录的功能, 会员用户只要在论坛登录页面中输入注册成功时的登录名和密码即可。单击按钮后可以检测用户的登录名和密码是否合法, 如果合法则可以进入页面进行其他功能的操作, 如果不合法则会提示重新登录。

3. 会员用户发帖

发帖即发表帖子, 只有登录成功的会员用户才享有对该功能的操作, 而未注册的用户 (即普通用户) 不能享有该功能。

4. 会员用户回帖

回帖即回复帖子, 登录成功的会员用户可以针

对某一领域的某个问题跟帖, 然后发表自己的意见、见解或看法。而普通用户不能实现回帖的功能。

5. 搜索或浏览帖子

普通用户和已注册的会员用户都享有浏览帖子和搜索帖子的功能, 浏览帖子即浏览不同领域和版块的所有帖子。他们也可以在搜索框中输入感兴趣的内容查看帖子列表, 然后单击查看其详细内容。

6. 新手手册

新手手册中的内容是由管理员负责管理的, 对于首次进入 BBS 论坛系统的会员或游客都可以通过查看新手手册来了解该系统的功能和使用。

7. 版块管理

版块管理是管理员和超级版主所特有的权限功能, 管理员可以对版块进行分类、删除版块、添加版块以及修改版块等。论坛提供了不同版块讨论区域的相关数量统计, 并且会员可以选择不同的版块区域进行讨论。

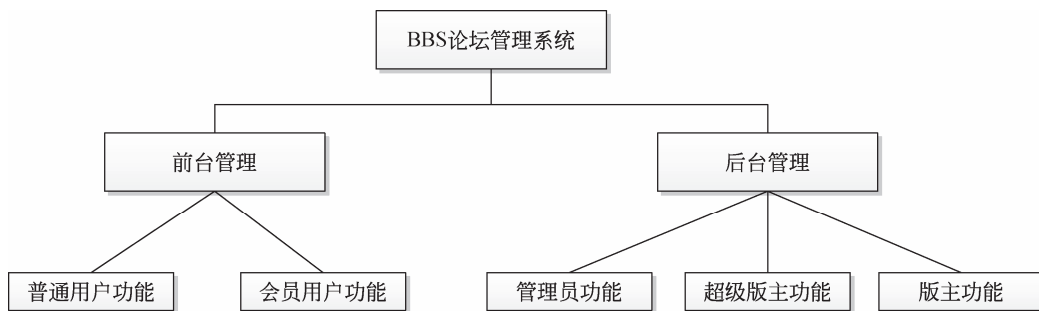
8. 帖子管理

管理员、超级版主和版主都可以对帖子进行管理, 如对帖子进行添加、删除、设置精华帖子以及控制点击率等操作。

9. 会员用户管理

管理员具有最高权限, 他可以对会员用户进行增加、删除、修改、查询以及将会员设置为版主等操作。用户添加完成后系统会把会员的相应资料添加到数据库中, 例如会员 ID、会员名称、会员密码、会员邮箱、会员联系电话和会员居住地址等。管理员会根据用户的身份进行相关内容的设置。将某个用户设置为版主后, 该会员用户可以对该版块下的帖子进行管理。

从上面的介绍中, 相信读者一定对 BBS 论坛的相关功能有所了解, 如下图所示为论坛系统总体的功能模块图。



从上图中可以看出，BBS论坛管理系统包括两部分：前台管理和后台管理。其中前台管理根据用户的身份可以划分为普通用户所享有的功能和注册成功的会员用户所享有的功能；后台管理则根据用户身份分别划分为管理员、超级版主和版主，身份不同所享有的功能也不完全相同。

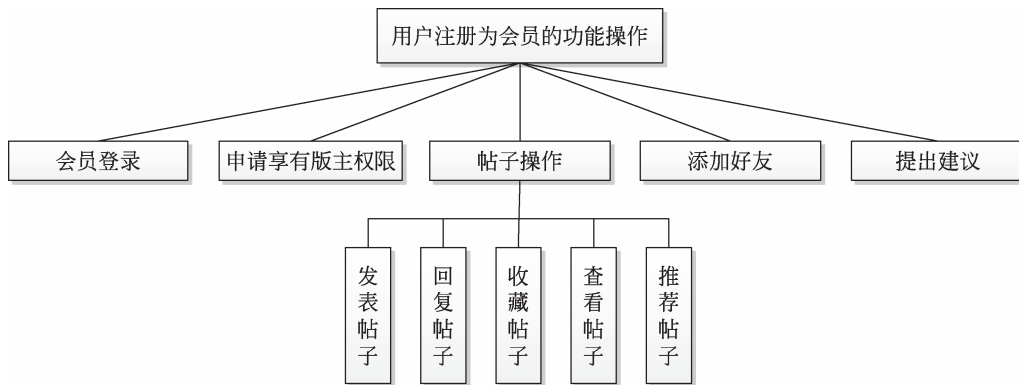
关操作，前台功能包括查看不同板块的帖子、根据条件搜索帖子、查看新帖、发表帖子、用户登录以及普通用户注册成为会员等操作。

1. 会员用户

由于用户的身份不同，所以他们所享有的功能权限也不相同，如下图所示演示了会员用户可以进行的功能操作。

3.4.2 前台功能概述

前台功能是指用户能够访问前台页面进行相



从上图中可以看出，会员用户主要包括5个功能操作：会员登录、申请享有版主权限、帖子操作、添加好友以及提出建议。其中帖子操作又包括发表帖子、回复帖子、收藏帖子、查看帖子以及将帖子设置为精华帖5个操作。

下面将简单介绍与会员用户相关功能的操作。

□ 会员登录

系统提供了会员登录功能，单击页面中的【登录】按钮，在登录页面输入注册成功的用户名和密码进行登录，只有验证成功后才能使用系统提供的功能。

□ 申请享有版主权限

登录成功的会员用户只享有普通会员的权限，每个会员的等级都可以进行提升，当会员升级到一定级数时就可以申请成为版主。版主可以对该区域内的帖子进行管理操作，如删除帖子和修改帖子等。

□ 添加好友

会员还可以将其他的会员添加为自己的好友，然后与好友分享自己发表、回复的帖子，同时还可以邀请好友欣赏自己收藏的帖子等。另外，会员也可以从好友列表中删除某个好友。

□ 提出建议

会员用户可以查看版主、超级版主和管理员所提出的建议，当然自己也可以向管理员或超级版主提出建议。

□ 发表帖子、回复帖子和查看帖子

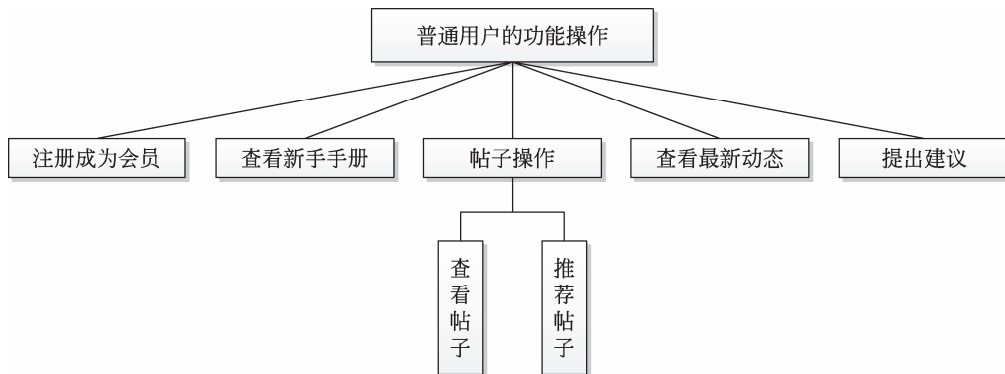
会员用户登录成功后可以对论坛中的帖子进行简单的基本操作，如会员可以在某个版块下发表帖子、对某个帖子进行回复或查看某个版块下帖子的详细内容等。

□ 收藏帖子和推荐帖子

论坛上的帖子有很多，有的甚至成百条、成千条，每次查找时也会相当麻烦，所以会员用户可以将自己喜欢的帖子进行收藏，这样方便以后查看。另外也可以选择特定的好友，将某个帖子推荐给他们。

2. 普通用户

前台除了为会员用户提供多个功能操作外，也为没有注册的普通用户提供了一些操作。如下图所示为普通用户的功能操作。



从上图中可以看出，普通用户的功能操作主要包括注册成为会员、查看新手手册、查看最新动态、提出建议以及帖子操作 5 个功能。其中帖子操作包含查看帖子和推荐帖子。

下面对普通用户的功能操作进行简单介绍。

□ 注册成为会员

BBS 论坛系统提供了对普通用户注册成为会员的功能，如果想要成为系统的会员，只要单击【用户注册】按钮，在注册页面输入个人信息（如用户名、密码、联系电话和性别等）即可。

□ 查看新手手册

普通用户进入 BBS 论坛系统后可以查看新手手册了解论坛的基本功能和操作步骤等，这样可以方便用户以最快的速度了解该论坛系统。

□ 查看最新动态

普通用户有权限了解当前论坛系统的最新动态，如发表的新帖子、新话题以及版本更新等内容。

□ 提出建议

会员具有向管理人员提出建议的功能，同样普通用户也有该功能权限。普通用户可以向会员、版

主或管理员等提出建议，当然也可以查看管理员向会员或普通用户所提出的建议和意见等。

□ 帖子操作

普通用户可以对论坛系统的帖子进行最基本的操作：查看帖子和推荐帖子。如果是未注册的用户（即普通用户），推荐帖子时不能够向指定的人进行推荐，而是向所有的会员进行推荐。

3.4.3 构造用例模型

用例图描述了一个外部的观察者对系统的印象，强调这个系统是什么，而不是这个系统怎么工作。在 BBS 论坛系统中，用例图的任务是明确系统是为哪些用户服务，即哪些用户需要利用 BBS 系统来工作。另外，还需要确定系统中的管理者和相关工作人员。

BBS 论坛系统中由于用户身份的不同，所涉及的用户功能也不相同。后台用户主要涉及管理员和版主，而前台用户主要涉及普通用户和会员。下面分别从会员用户和普通用户两方面绘制功能用例图。

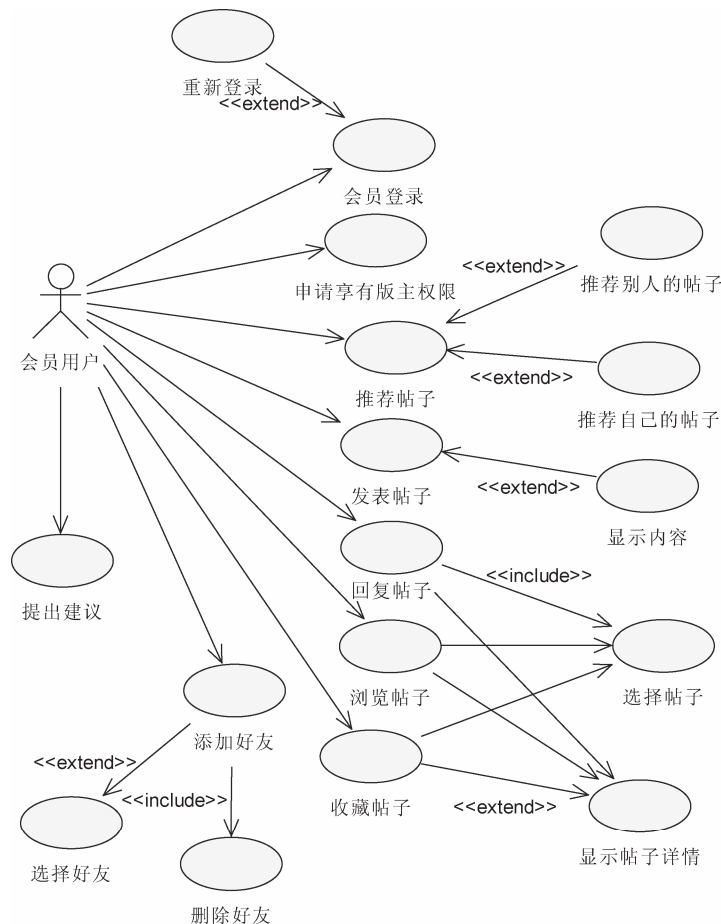
1. 会员用户功能用例图

用例图的构成包括系统、参与者、用例和关系（如泛化关系、包含关系和扩展关系）。创建用例图模型的基本步骤如下。

- (1) 确定系统涉及的总体信息。
- (2) 确定系统的参与者。
- (3) 确定系统的用例。

(4) 构造用例模型。

上文中的会员用户图中已经显示了与会员用户相关的功能操作。在与会员相关的用例图中涉及会员用户、会员要操作的会员登录、推荐帖子、发表帖子、回复帖子以及浏览帖子等功能操作。根据上面的操作步骤绘制会员用户功能的用例图，如下图所示。

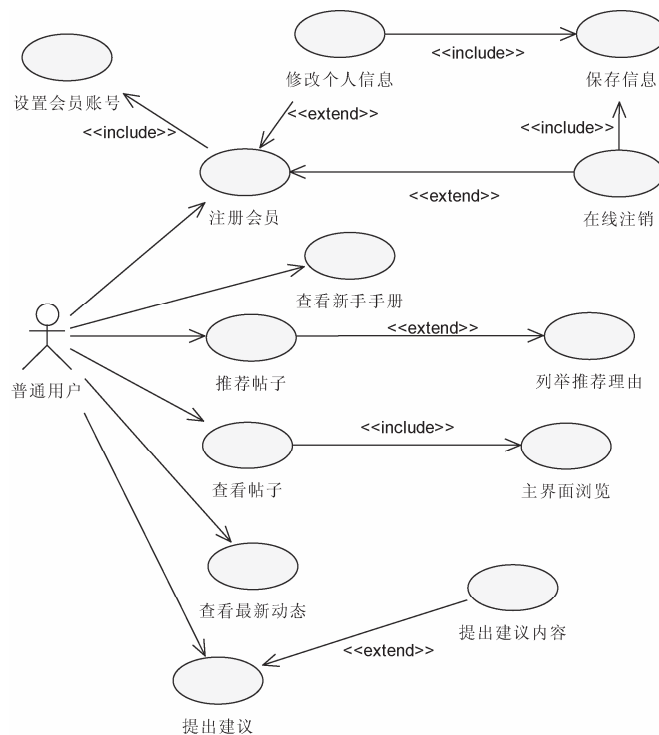


上图中包含会员的多个功能操作，如下是对会员主要功能用例的分析。

- 会员可以选择帖子查看帖子详情，并且对某个帖子进行回复、浏览和收藏等。
- 会员可以向管理员发送请求成为版主的要求。
- 会员可以选择添加好友，并且可以删除好友。

2. 普通用户功能用例图

除了会员操作外，上文中的普通用户图中也列出了普通用户常用的功能操作。例如，普通用户可以注册成为会员，注册成功后可以修改个人信息，也可以注销当前登录；普通用户可以将自己认为好的帖子向所有人进行推荐，也可以向所有的版主和管理员发送建议等。根据绘制用例图的步骤绘制普通用户功能用例图，如下图所示。

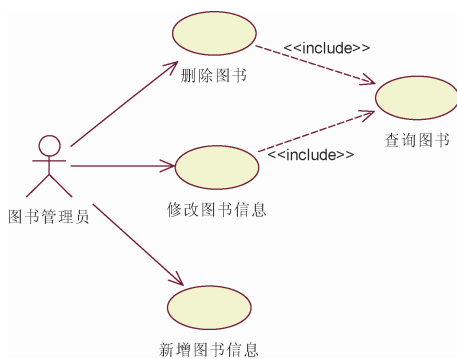


3.6 新手训练营

练习 1：图书管理用例图

downloads\3\新手训练营\图书管理用例图

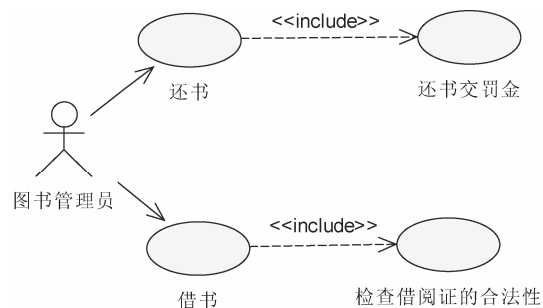
提示：本练习中，将创建一个图书管理用例图。图书馆中的图书根据需求进行更新是一项日常业务，管理员成功登录图书管理系统的书籍信息管理系统，对图书进行新增图书信息、删除图书、修改图书信息，以及查询图书等操作。因此，在该用例中只包含一个图书管理员角色，3 个用例和 2 个依赖关系。



练习 2：图书借阅和归还用例图

downloads\3\新手训练营\图书借阅和归还用例图

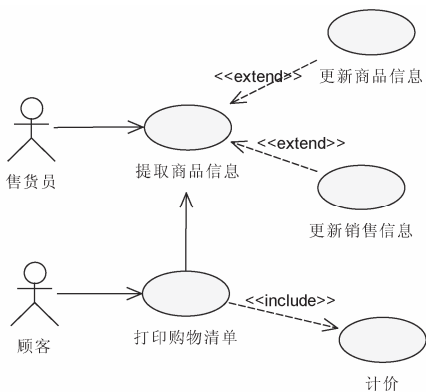
提示：本练习中，将创建一个图书借阅和归还用例图。从图书馆借书，是图书馆提供的一项基本服务。读者通过系统验证后，成功登录系统进行图书的借阅和归还。在该用例图中，只有图书管理员一个角色，包含还书、借书 2 个用例和 2 个依赖关系。



练习 3：销售管理子系统用例图

downloads\3\新手训练营\销售管理子系统用例图

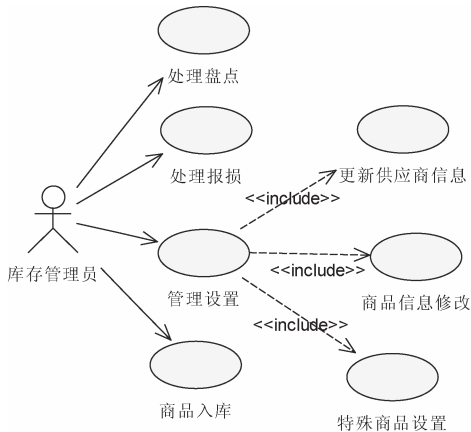
提示：本练习中，将创建一个超市销售管理子系统用例图。销售管理子系统中主要包含了售货员和顾客 2 个角色，其中售货员可以提取商品信息，包括更新商品信息和更新销售信息；而顾客则可以浏览商品信息和打印购物清单等。



练习 4：库存管理子系统用例图

downloads\3\新手训练营\库存管理子系统用例图

提示：本练习中，将创建一个库存管理子系统用例图。库存管理子系统主要涉及库存管理中的各项工作，包含了库存管理员一个角色，该角色可以执行管理设置、商品入库、处理盘点、处理报损等一系列的功能。

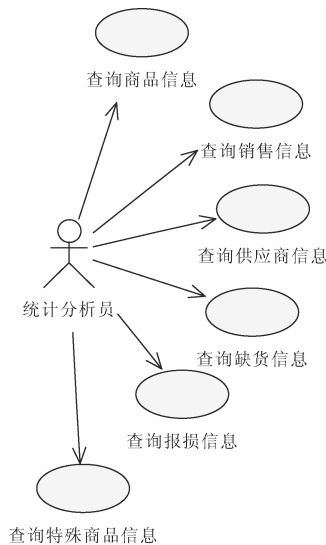


练习 5：统计分析子系统用例图

downloads\3\新手训练营\统计分析子系统用例图

提示：本练习中，将创建一个统计分析子系统用例图。统计分析子系统是统计分析员对商品进行一系列的分析操作，该用例中只包含统计分析员一个角色，包含了查询商品信息、查询销售信息、查询供应商信息、查询缺货信息、查询报损信息、查询特殊商品信息等用例。

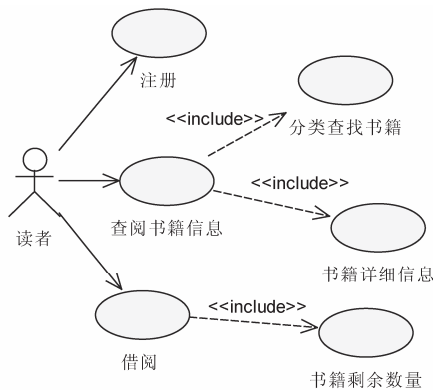
色，包含了查询商品信息、查询销售信息、查询供应商信息、查询缺货信息、查询报损信息、查询特殊商品信息等用例。



练习 6：读者信息管理系统用例图

downloads\3\新手训练营\读者信息管理系统用例图

提示：本练习中，将创建一个读者信息管理系统用例图。读者信息管理系统主要是读者在图书系统中进行注册、查阅和借阅等一系列的操作。该用例中，包含了读者一个角色，同时包含了注册、查阅书籍信息、借阅等用例。



练习 7：系统后台用例图

downloads\3\新手训练营\系统后台用例图

提示：本练习中，将创建一个系统后台用例图。系统后台用例图主要展示了系统管理员对图书管理员信息、登录、图书分类管理等用例的执行情况。在

该用例中，只包含系统管理员一个角色，包含了图书管理员信息管理、登录和图书分类管理 3 个用例，还包括添加管理员、删除管理员、添加图书分类、删除图书分类和修改图书分类 5 个泛化关系，以及查看管理员操作记录一个依赖关系。

