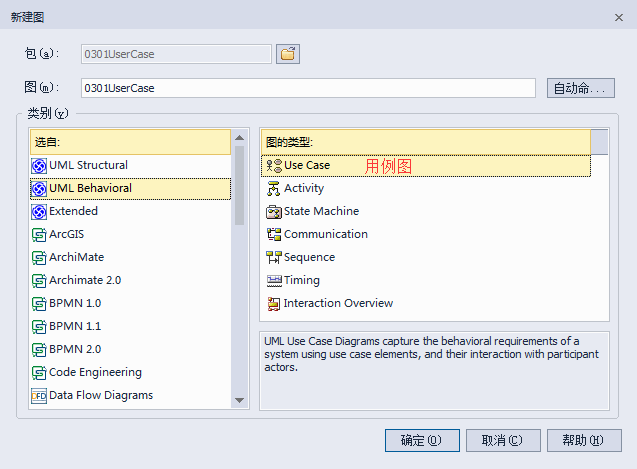
**[UML建模—EA创建Use Case(用例图)](https://www.cnblogs.com/sxhlf/p/6830507.html)**

用例图主要用来描述“用户、需求、系统功能单元”之间的关系。它展示了一个外部用户能够观察到的系统功能模型图。

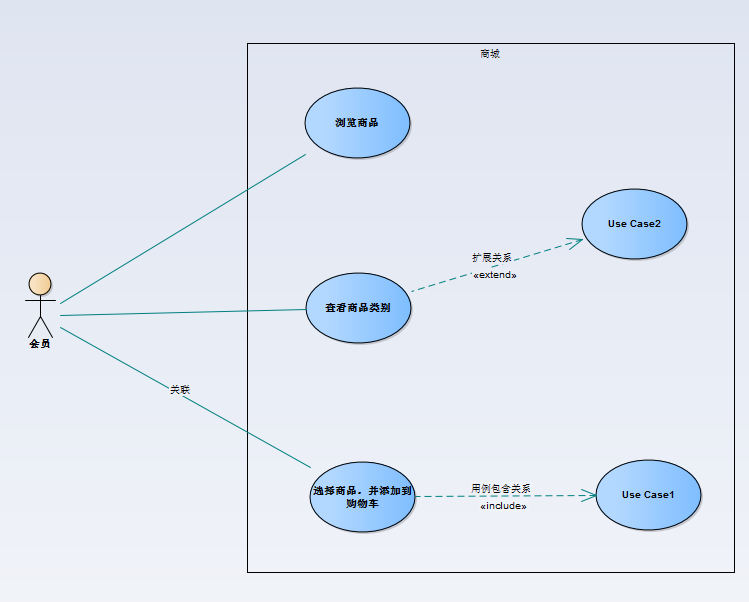
1.新建用例图



2.用例图工具：



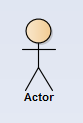
3.一个简单用例：



用例图所包含的元素如下：

**1. Actor：参与者**

    表示与您的应用程序或系统进行交互的用户、组织或外部系统。用一个小人表示。



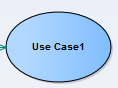
使用者是系统的一个用户 ；用户可以意味着人类的用户，一台机器，或甚至另一个系统或子系统模型中。任何从外面或系统边界与系统交互的东西被称为一个使用者。使用者通常与用例相关联。

使用者可以使用系统，通过图形用户界面、 批处理接口或一些其他媒体。在用例的情况下，系统必须提供功能的详细信息记录和一个使用案例与使用者的互动来满足用户的要求。

使用者也代表了在序列图中，可以显示这些用户的角色并使用矩形表示法。Enterprise Architect 业务建模支持构造型的使用者元素。业务建模元素还将使用者表达作为构造型对象。

**2. 用例(Use Case)**

　　用例就是外部可见的系统功能，对系统提供的服务进行描述。用椭圆表示。



    用例 是 一UML 建模元素，用来描述一个拟议系统的用户是如何与系统交互的，以执行离散的功能单元。它描述，并表示为最终用户 （人员、 机器或其它系统） 的意义，须离开系统中完成状态的时间单相互作用：

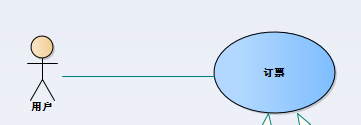
互动已完成或回滚到用例初始状态：

* 通常具有要求和约束，描述用例的基本特征和规则
* 可以有相关联的序列图说明行为的随时间的变化，谁给谁在做什么，何时做的
* 通常情况下有场景与之相关联， 描述工作流随时间产生的结果。已关联可选择的工作流 （例如，要捕的异常） 也会启用

**3. 关系**

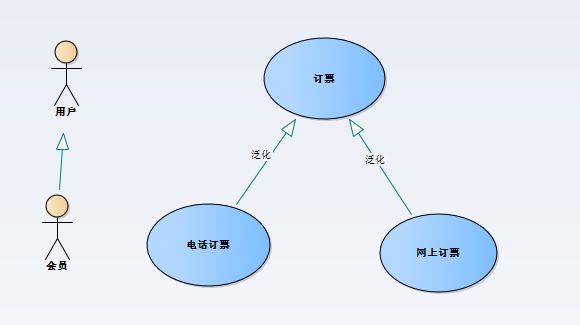
　　用例图中涉及的关系有：关联、泛化、包含、扩展。

**a. 关联(Associate)**

****

　  　表示参与者与用例之间的通信，任何一方都可发送或接受消息。

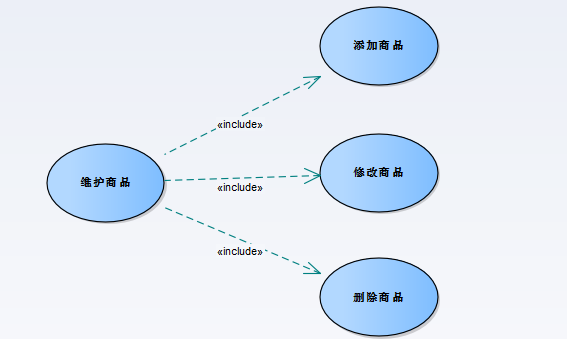
**b. 泛化(Generalize)**

****

       就是通常理解的继承关系，子用例和父用例相似，但表现出更特别的行为；子用例将继承父用例的所有结构、行为和关系。子用例可以使用父用例的一段行为，也可以重载它。父用例通常是抽象的。

　　【箭头指向】：指向父用例

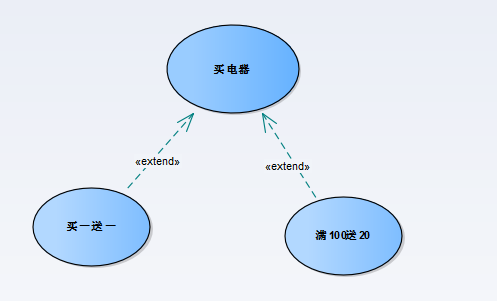
**c. 包含(Include)**

****

       包含关系用来把一个较复杂用例所表示的功能分解成较小的步骤。

　　【箭头指向】：指向分解出来的功能用例

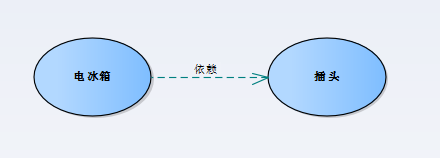
**d. 扩展(Extend)**

****

　　扩展关系是指用例功能的延伸，相当于为基础用例提供一个附加功能。

　　【箭头指向】：指向基础用例

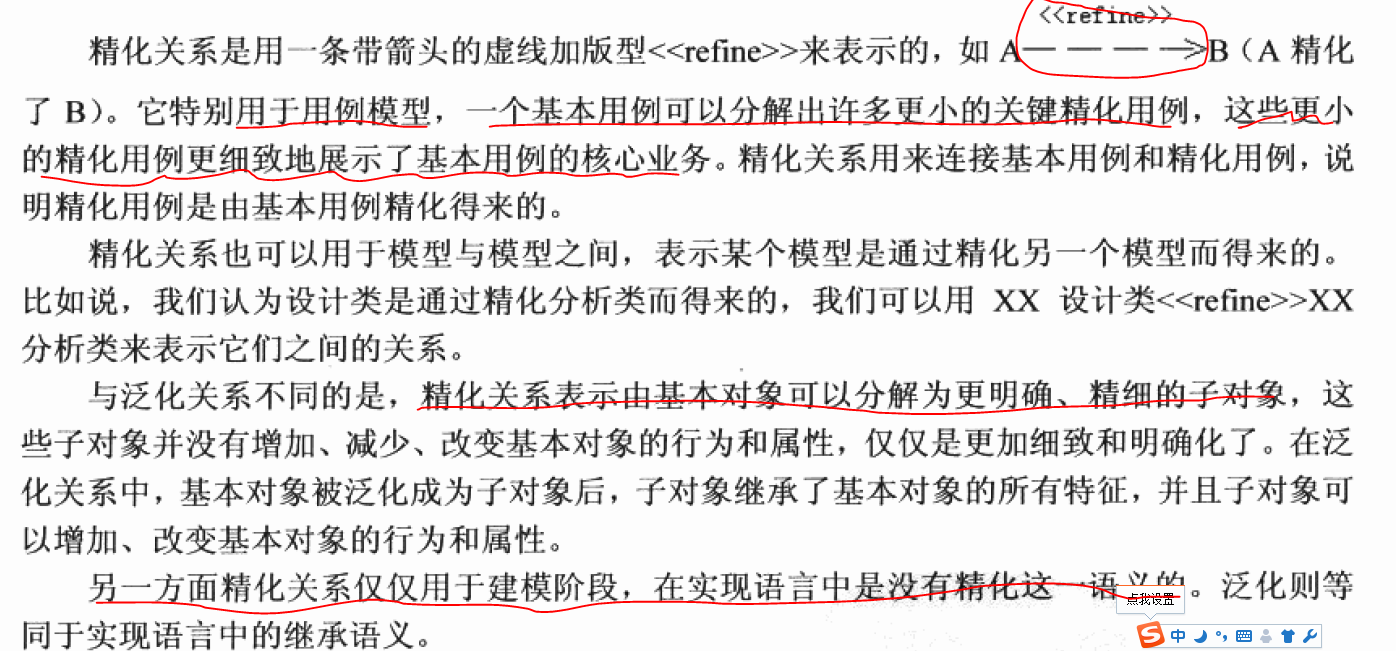
**e. 依赖(Dependency)**

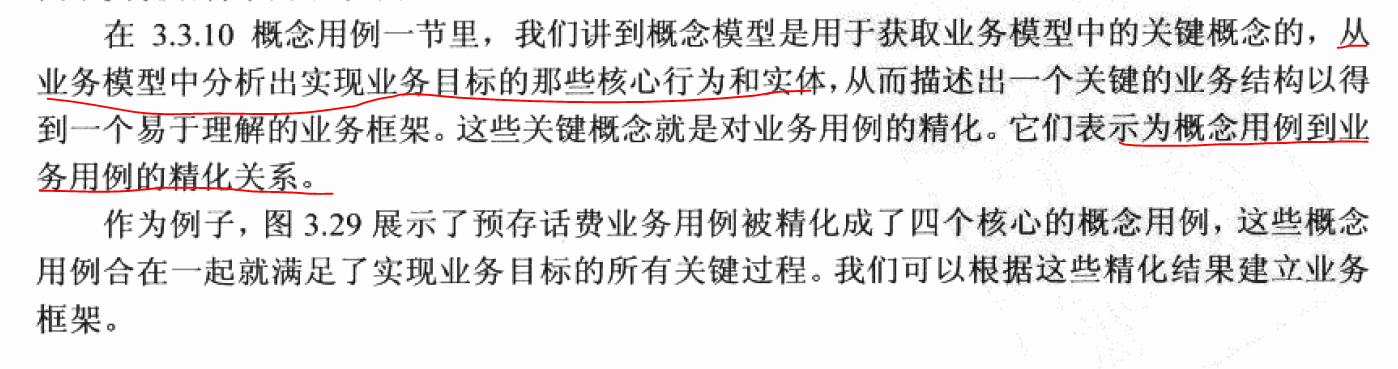
****

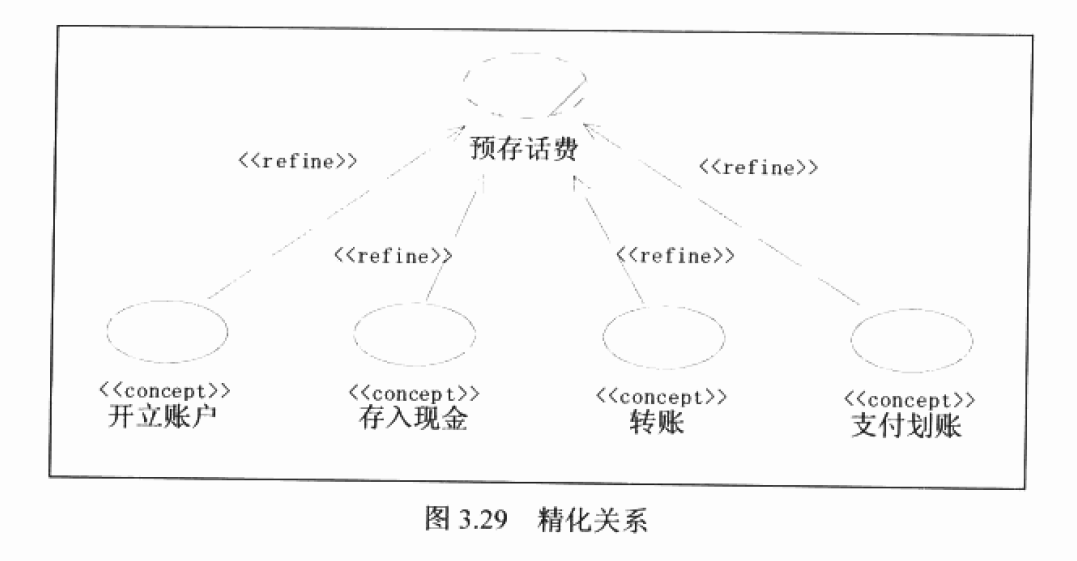
　　以上4种关系，是UML定义的标准关系。但VS2010的用例模型图中，添加了依赖关系，用带箭头的虚线表示，表示源用例依赖于目标用例。

【箭头指向】：指向被依赖项

**f.精化关系（refine）**







**4. 包含(include)、扩展(extend)、泛化(Generalize)** **的区别：**

　　条件性：泛化中的子用例和include中的被包含的用例会无条件发生，而extend中的延伸用例的发生是有条件的；

　　直接性：泛化中的子用例和extend中的延伸用例为参与者提供直接服务，而include中被包含的用例为参与者提供间接服务。

　　对extend而言，延伸用例并不包含基础用例的内容，基础用例也不包含延伸用例的内容。

　　对Inheritance而言，子用例包含基础用例的所有内容及其和其他用例或参与者之间的关系；