

# UML和需求分析

---

## UML和需求分析

回顾

教学目标

### 第一章 UML简介

1.1 什么是UML

1.2 UML的目的

1.3 UML图的分类以及作用

1.3.1 UML 组件图

1.3.2 UML 用例图

1.3.3 UML 时序图（顺序图）

1.3.4 UML 活动图（流程图）

1.4 软件的各个开发阶段需要的UML图

### 第二章 Enterprise Architect的介绍

### 第三章 常用UML图的使用

3.1 UML用例图

3.1.1 怎么画用例图

3.1.2 用EA画用例图（老师演示例子）

3.1.3 学生做题

3.2 UML组件图

3.2.1 怎么画组件图

3.2.2 用EA画时序图（老师演示例子）

3.2.3 学生做题

3.3 UML时序图

3.3.1 怎么画时序图

3.3.2 用EA画时序图（老师演示例子）

3.3.3 学生做题

3.4 UML活动图（流程图）

3.4.1 怎么画活动图

3.4.2 用EA画活动图（老师演示例子）

3.4.3 学生做题

### 第四章 随堂项目背景介绍

### 第五章 开放平台设计说明

### 第六章 学生课后作业

### 第七 面试题

## 回顾

---

昨天作业讲解、知识回顾。

## 教学目标

---

- 1、了解UML的作用
- 2、能看懂常见的UML图
- 3、可以用Enterprise Architecture软件画常用的UML图（用例图、组件图、时序图、活动图）
- 4、开放平台背景介绍与项目的需求分析

# 第一章 UML简介

## 1.1 什么是UML

UML 是 OMG 在1997年1月提出了创建由对象管理组和 UML1.0 规范草案；

UML 是一种为面向对象开发系统的产品进行说明、可视化、和编制文档的标准语言；

UML 作为一种模型语言，它使开发人员专注于建立产品的模型和结构，而不是选用什么程序语言和算法实现；

UML 是不同于其他常见的编程语言，如C++，Java中，COBOL等，它是一种绘画语言，用来做软件蓝图；

UML 不是一种编程语言，但工具可用于生成各种语言的代码中使用UML图；

UML 可以用来建模非软件系统的处理流程，以及像在一个制造单元等。

## 1.2 UML的目的

UML 的目标是定义一些通用的建模语言并对这些建模语言做出简单的说明，这样可以让建模者理解与使用。UML 也是为普通人和有兴趣的人而开发的系统，它可以是一个软件或者使用非软件，它必须是明确的。我们不将 UML 作为一个开发方法，而是随着流程做一个成功的系统。

现在我们可以明确的了解 UML 的目标就是 UML 被定义为一个简单的建模机制，帮助我们按照实际情况或者按照我们需要的样式对系统进行可视化；提供一种详细说明系统的结构或行为的方法；给出一个指导系统构造的模板；对我们所做出的决策进行文档化。

UML的目标之一就是为开发团队提供标准通用的设计语言来开发和构建计算机应用。UML提出了一套IT专业人员期待多年的统一的标准建模符号。通过使用UML，这些人员能够阅读和交流系统架构和设计规划--就像建筑工人多年来所使用的建筑设计图一样。

简单地说，就像建设计师与建筑工人交流的语言是设计图一样，软件架构师或设计师与程序员交流沟通的图就是UML。

## 1.3 UML图的分类以及作用

UML 的核心是图表，大致可以将这些图归类为结构图和行为图。结构图是由静态图，静态图分为：用例图，类图，对象图，包图，构件图，部署图。行为图分为：状态图，活动图，协作图，序列图。

我们将主要介绍以下四种常用图的使用：

### 1.3.1 UML 组件图

组件图即是用来描述组件与组件之间关系的一种UML图。组件图在宏观层面上显示了构成系统某一个特定方面的实现结构。组件图中主要包含三种元素，即组件、接口和关系。

组件图通过这些元素描述了系统的各个组件及之间的依赖关系，还有组件的接口及调用关系。

组件图在面向对象设计过程中起着非常重要的作用：它明确了系统设计，降低了沟通成本，而且按照面向对象方法进行设计的系统和子系统通常保证了低耦合度，提高了可重用性。

### 1.3.2 UML 用例图

"用例图"用来描述谁用系统，用系统做什么。用例图不涉及使用细节，只用来描述使用人员和系统的关系，也不涉及行动的顺序。用例图是指由参与者（Actor）、用例（Use Case），边界以及它们之间的关系构成的用于描述系统功能的视图。用例图（User Case）是外部用户（被称为参与者）所能观察到的系统功能的模型图。

### 1.3.3 UML 时序图（顺序图）

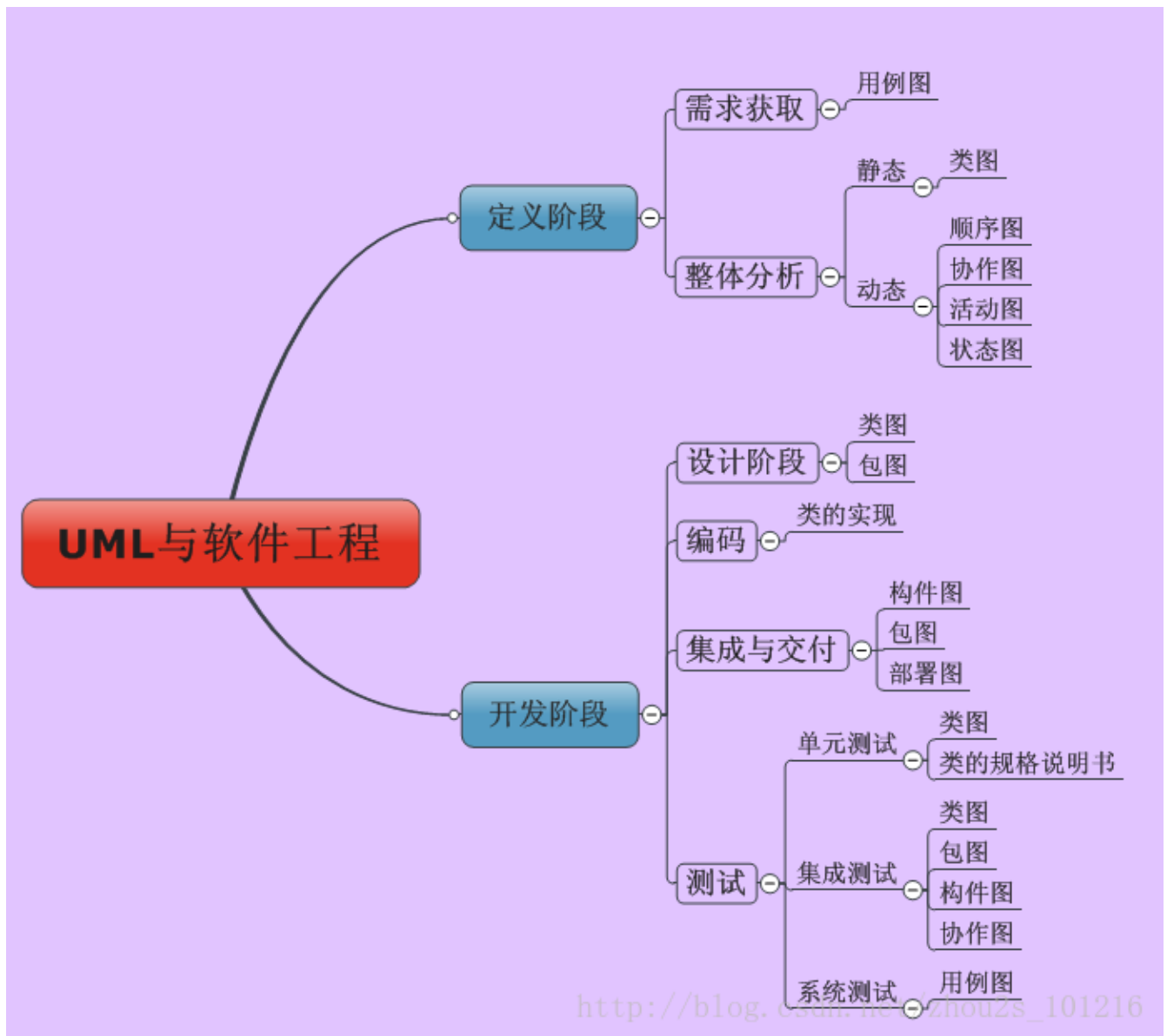
是一种UML交互图。它通过描述对象之间发送消息的时间顺序显示多个对象之间的动态协作。换句话说就是描述谁和谁先做什么，然后再做什么的图。时序图中包括的建模元素主要有：对象（Actor）、生命线（Lifeline）、控制焦点（Focus of control）、消息（Message）等等。

### 1.3.4 UML 活动图（流程图）

活动图是UML用于对系统的动态行为建模的另一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。

活动图用来描述一个操作的执行过程中所完成的工作或者动作；描述对象内部的工作，显示如何执行一组相关的动作，以及这些动作如何影响周围对象；描述用例的执行。

## 1.4 软件各个开发阶段需要的UML图



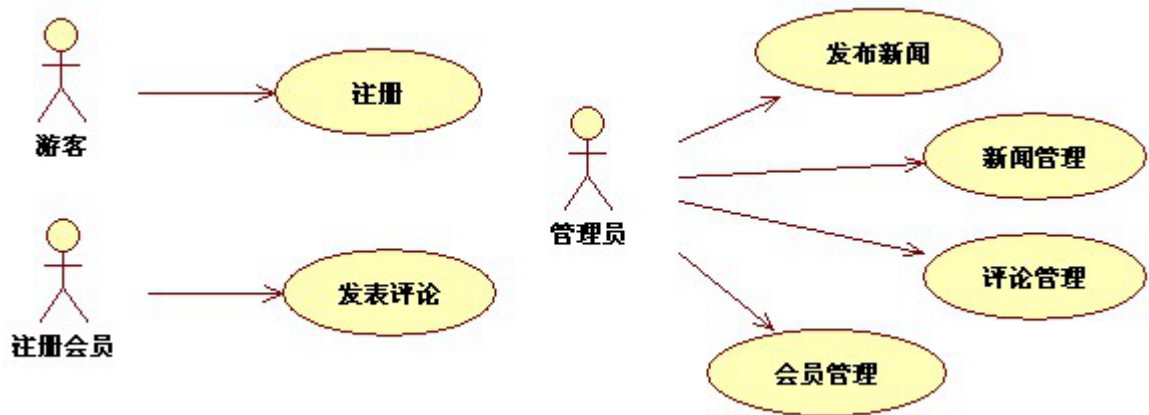
UML有这么多图，具体怎么用呢？

### 一、需求分析阶段的业务用例图

用例图，是用来表示系统角色与系统什么功能发生交互的图。通过用例图，可以很清晰地表示系统放主要功能。用例图在我们进行软件分析阶段和设计阶段都有使用：

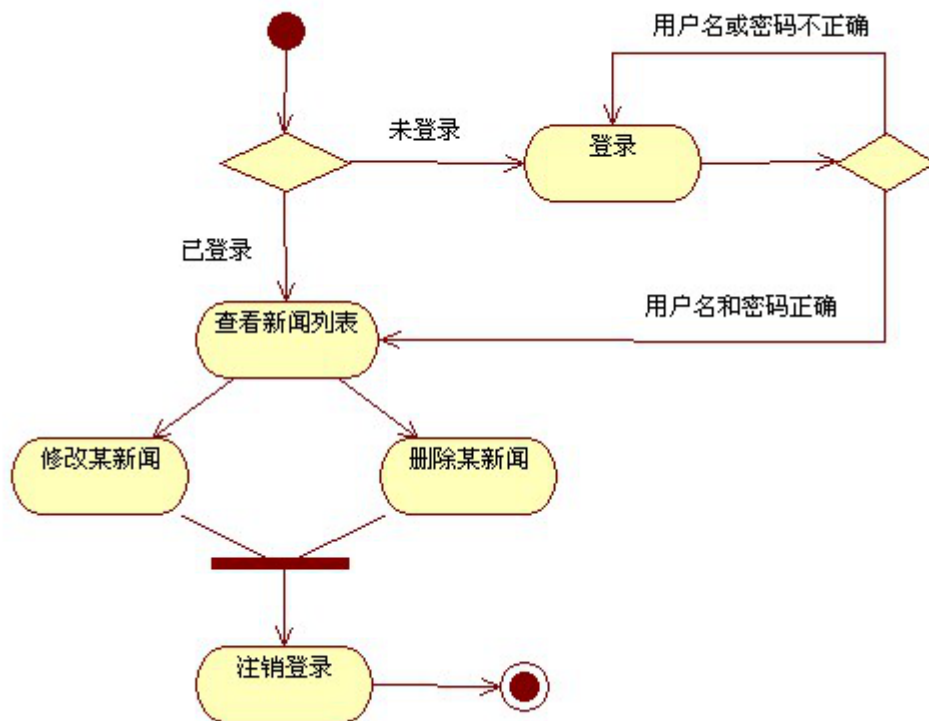
由用户需求得到业务用例（描述最主要的业务功能，客户最感兴趣的、期望的功能）

在与客户第一次交流沟通，采集需求后。我们可以得到《开发文档1.0》。同时，也可以由客户描述的系统功能、用户角色画出业务用例图。注意：这只是初步的用例，用来说明系统业务功能的。例如：一个新闻网站的业务用例图如下：

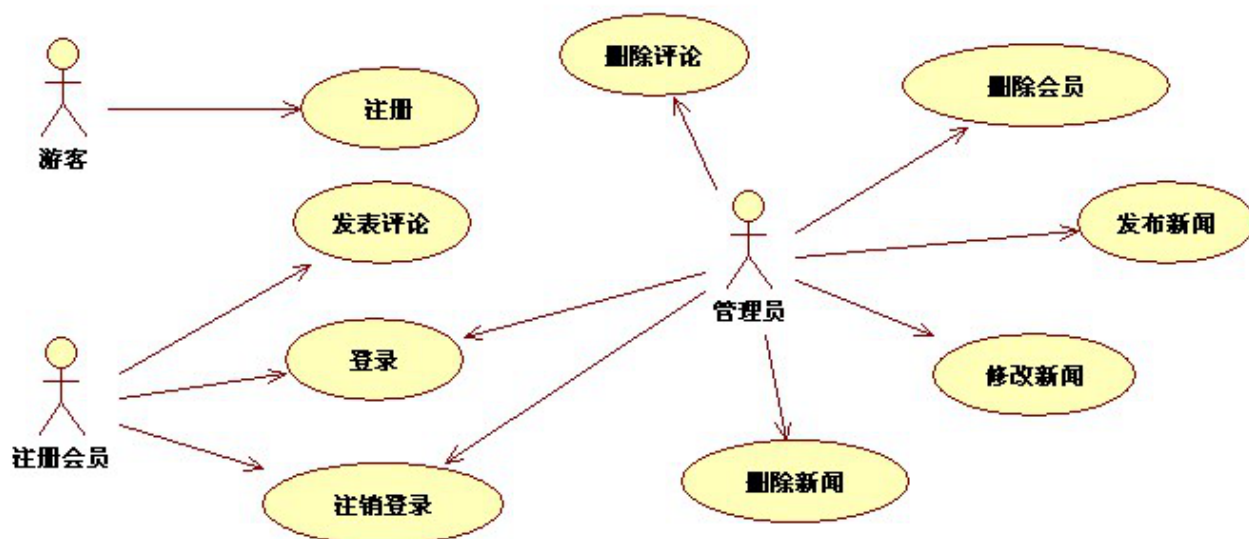


## 二、概要设计阶段的功能活动图、系统用例图

1、在把《开发文档1.0》和业务用例图交予客户审核确认后，我们开始编写《开发文档2.0》，然后根据《开发文档2.0》中新增的功能描述，我们可以画出每一个功能的活动图：例如：管理员原理新闻的功能活动图



2、由每一个功能活动图，完善业务用例图得到系统用例图（此时才是真正全面描述系统各个角色可以执行什么功能的用例图）



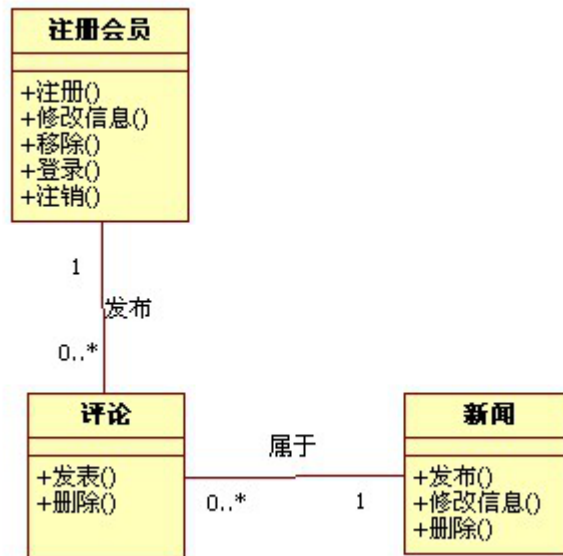
### 三、详细设计阶段的用例规约图

由《开发文档3.0》中的“功能详细设计”部分，画出每一个功能用例的约束图，主要包括：用例名、用例流程、异常处理等操作

用例名称	登录系统
用例简述	用户登录 CMS 系统
用例图	<pre> graph LR     User((用户)) --&gt; Login([登录])   </pre>
主要流程	1) 用户输入用户名和密码 2) 选择用户类型 3) 点击登录按钮
替代流程	3a) “用户名或密码错误”，系统出现用户名或密码错误的提示信息，回到主要流程 1，由用户重新输入 3b) “输入的用户名与类型不符”，系统出现提示信息，回到主要流程 1，由用户重新输入 3b) 当用户点击取消按钮时，取消登录

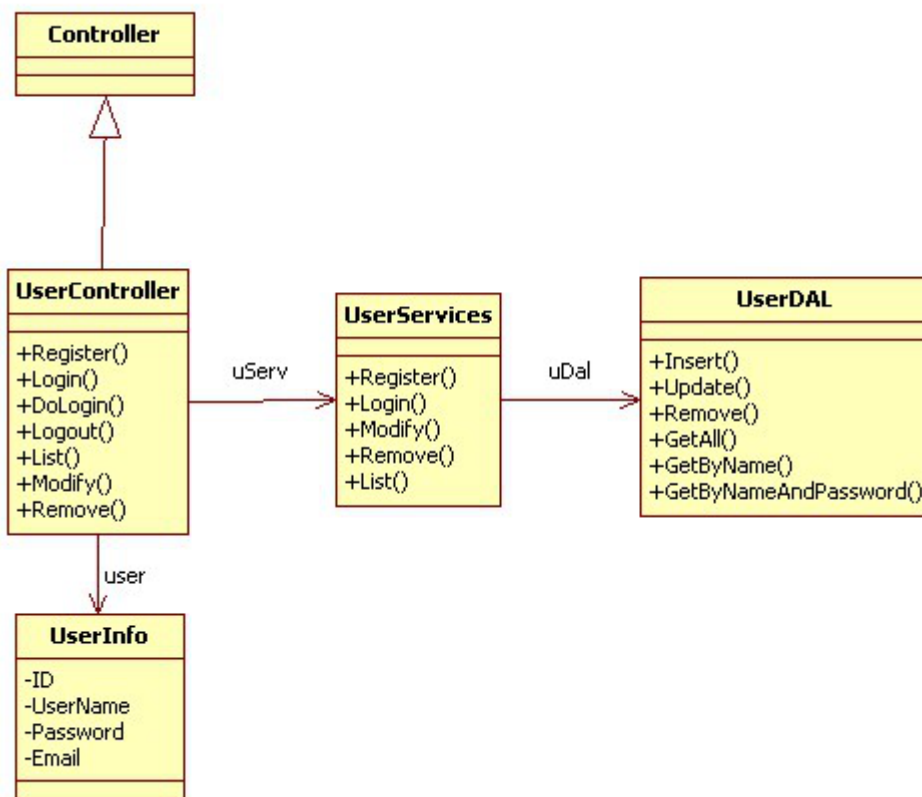
### 四、详细设计阶段的业务模块图

根据《开发文档4.0》中的“模块划分”，我们就知道了这个系统主要会有哪些业务类，画出业务模块图，每个业务类下列该模块下的功能用例：



## 五、详细设计阶段的类图

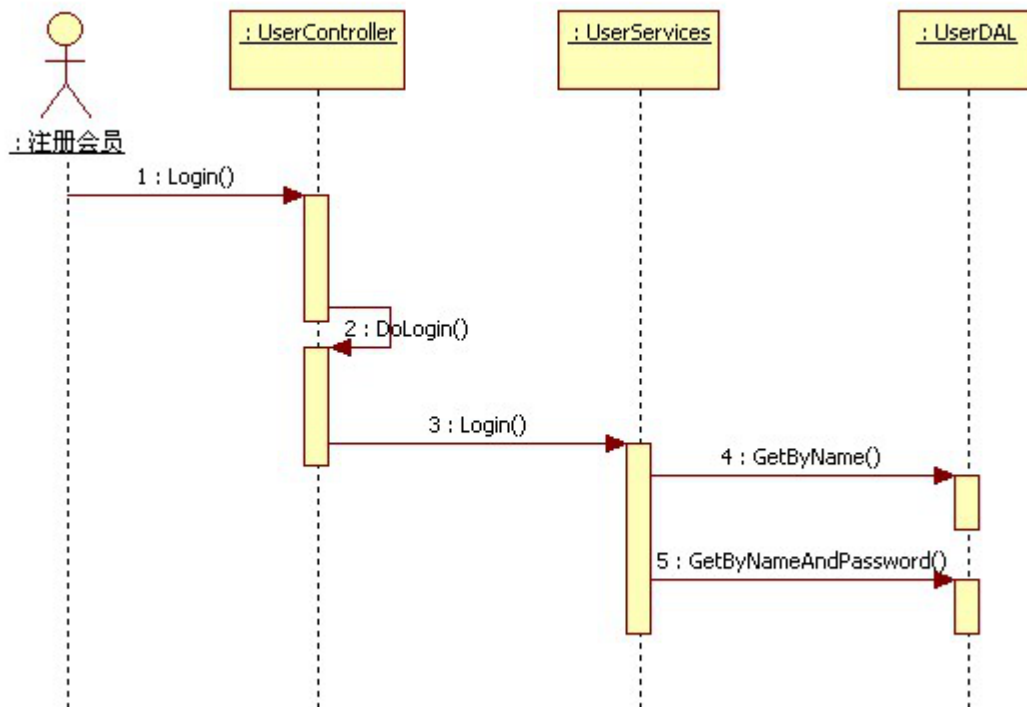
根据《开发文档5.0》中对每个用例的架构、以及功能模块的划分，可以初步确定系统需要多少个实现类组成，画出类图：



## 六：详细设计阶段的时序图

根据每个用例的活动图以及第五步的系统类图，我们可以为每个用例画出时序图，更加清晰明确地模拟出用户是怎么一步步调用哪个类的哪个方法来实现进行功能交互的，如：





七、根据上面的类图、用例的时序图等等就可以很清晰明确地进行各个类、模块的编码开发工作了

## 第二章 Enterprise Architect的介绍

1.EA(EnterpriseArchitect)是一个对于软件系统开发有着极好支持的CASE软件（Computer Aided Software Engineering）。EA不同于普通的UML画图工具（如VISIO），它将支撑系统开发的全过程。在需求分析阶段，系统分析与设计阶段，系统开发及部署等方面有着强大的支持，同时加上对10种编程语言的正反向工程，项目管理，文档生成，数据建模等方面。可以让系统开发中各个角色都获得最好的开发效率。2.生命周期软件设计方案——Enterprise Architect是以目标为导向的软件系统。它覆盖了系统开发的整个周期，除了开发类模型之外，还包括事务进程分析，使用案例需求，动态模型，组件和布局，系统管理，非功能需求，用户界面设计，测试和维护等。

### 主要用途

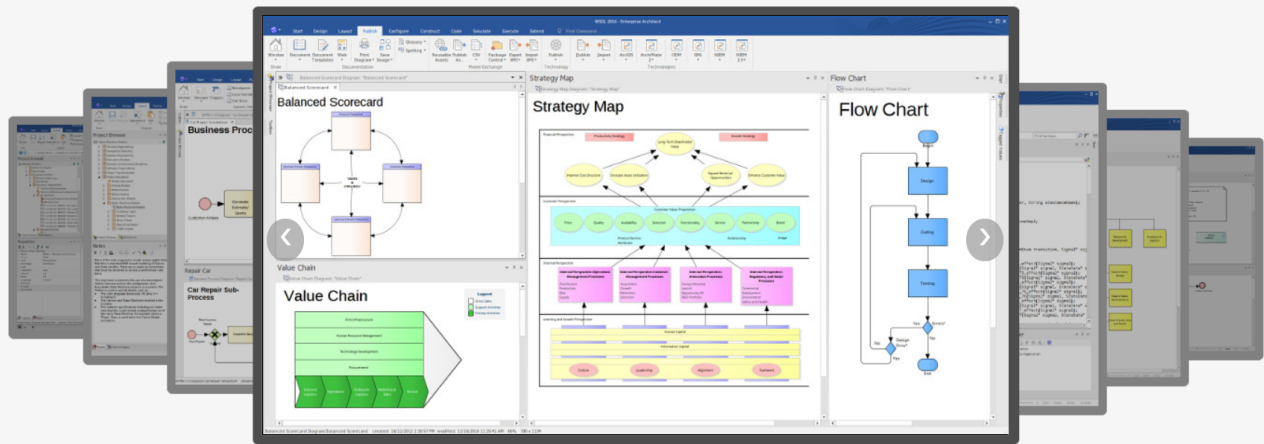
近20年来，Enterprise Architect一直是各种项目类型和工作区域的首选工具，支持这些领域中的大多数工作角色和功能。Enterprise Architect非常适合：

- 战略建模
- 需求开发和管理
- 业务流程建模和分析
- 软件开发
- 系统工程和数据库工程
- 企业架构
- 测试和调试
- 以及更多



## 快速直观的建模与设计

完美的企业级解决方案，可视化，分析，建模，测试和维护您的所有系统，软件，流程和架构。Enterprise Architect是一个理想的平台，可帮助您控制工作空间，支持您的同事和团队，实现协作并在最复杂的项目中建立信心。



# 第三章 常用UML图的使用

我们重点针对工作中常用的UML图来讲解。

## 3.1 UML用例图

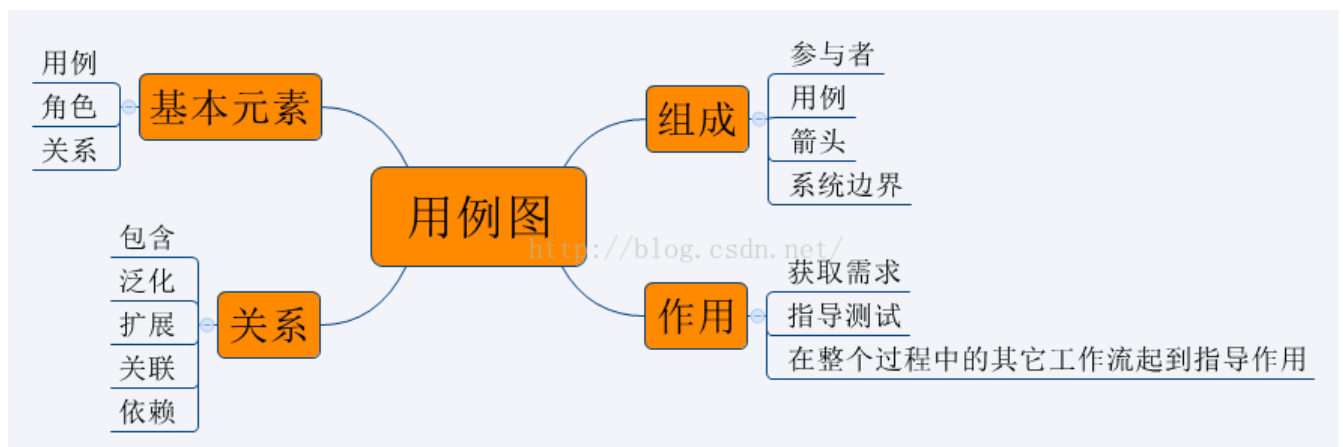
### 3.1.1 怎么画用例图

#### 定义

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静态视图。用例图（User Case）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，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。

用例图的组成：系统边界、参与者、用例、关系。

参与者：Actor不是人，而是指参与用例时担当的角色。



#### 怎样识别参与者呢？

1. 是谁向系统提供的信息呢。

2. 谁向系统获取信息。
3. 谁操作系统。

### 怎样确定用例的粒度呢？

用例的粒度（用例的大小）可大可小，一般一个系统易控制在20个左右。用例是系统级的抽象的描述，不是细化的（是做什么，非怎样做）。对复杂系统可以划分为若干个子系统处理。

### 怎样获取用例呢？

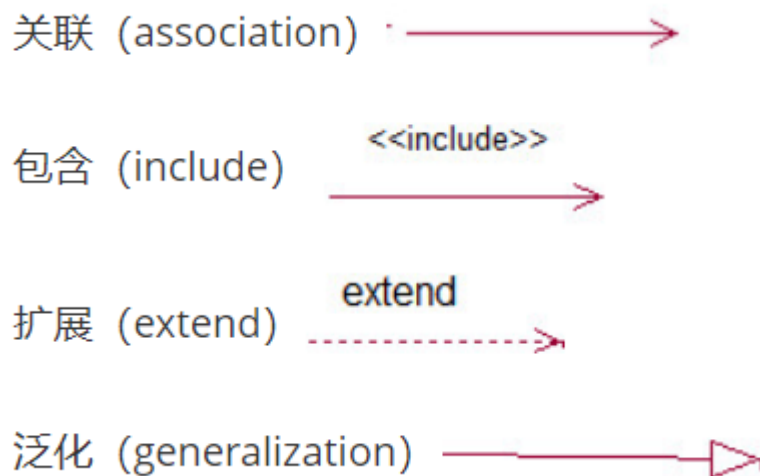
参与者希望系统执行什么任务？

参与者在系统中访问哪些信息（创建、存储、修改、删除等）？

需要将外界的哪些信息提供给系统？

需要将系统的那个事件告诉参与者？

### UML中的四种关系。



### 关联关系

描述参与者和用例之间的关系。

用单向箭头，表示谁启动用例。

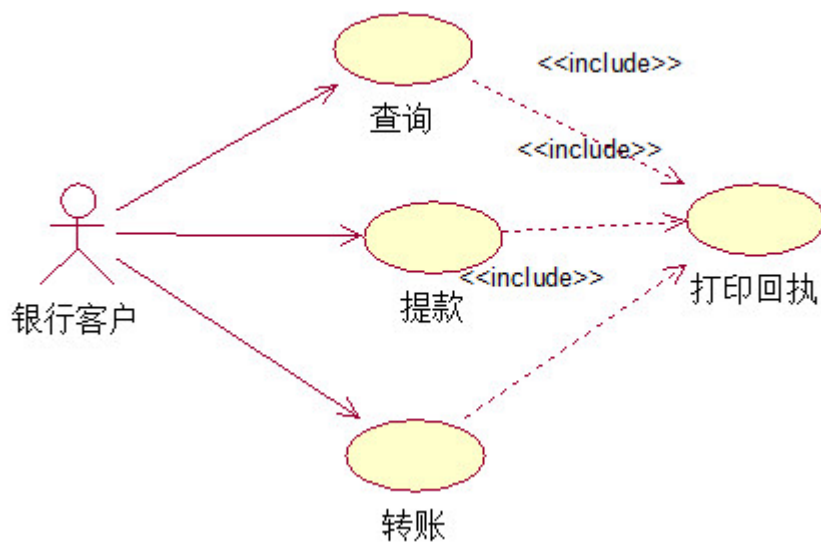
每个用例都有角色启动，除了包含和扩展用例。



### 包含

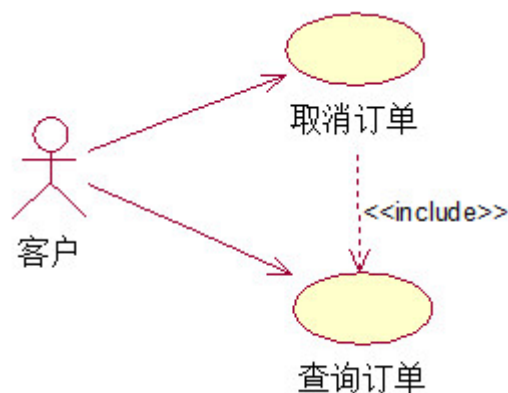
是指两个用例之间的关系。其中一个用例（基本用例，base use case）的行为包含了另一个用例（包含用例，inclusion use case）的行为。

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。

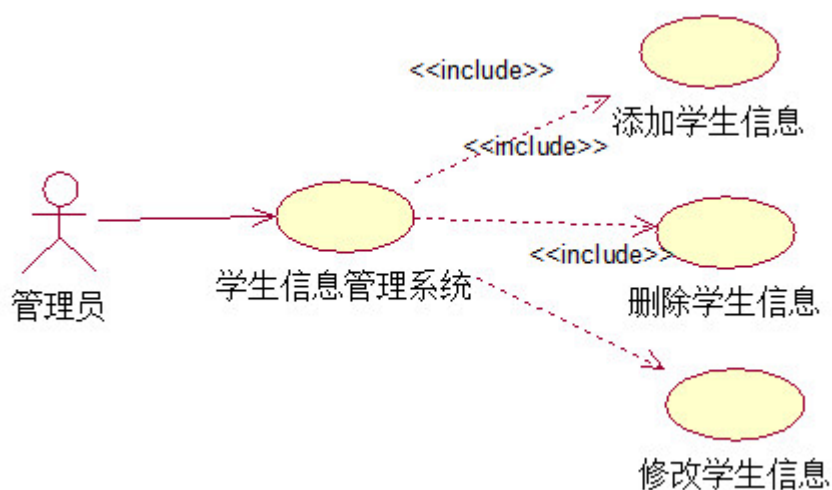


上面的例子就是说查询、提款和转账三个用例都有一个一致的功能，所以将这个功能提取出来为一个用例。且这三个用例和提取出的这个用例之间是包含的关系。

执行基本用例的时候也可以执行被包含的用例，被包含的用例也可以单独执行。



如果一个用例的功能太多时，可以用包含关系建模成两个或多个小用例



扩展。

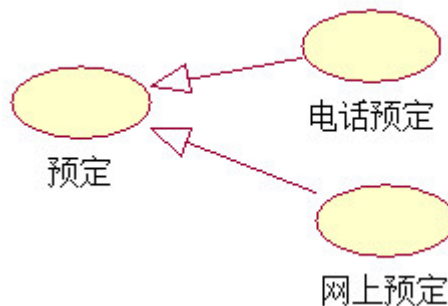
也是指**两个用例之间**的关系。一个用例可以被定义为基础用例的增量的扩展，称作为扩展关系。扩展关系是把新的行为插入到已有的用例中方法。基础用例即使没有扩展用例的执行不会涉及扩展用例，只有在特定的条件发生，扩展用例才被执行。



**泛化（继承）。**

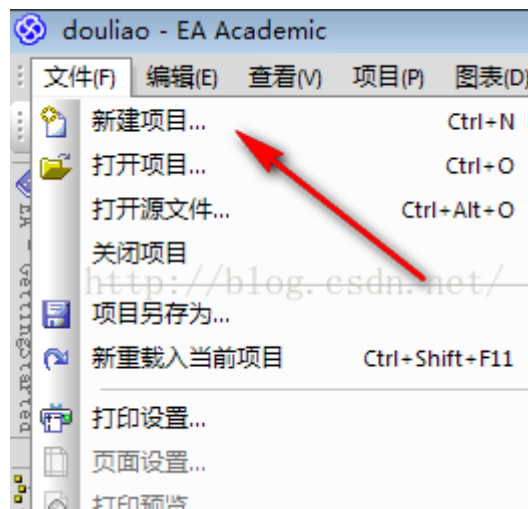
一个用例和其几种情形的用例间构成泛化关系。往往父用例表示为抽象用例。

任何父用例出现的地方子用例也可出现。

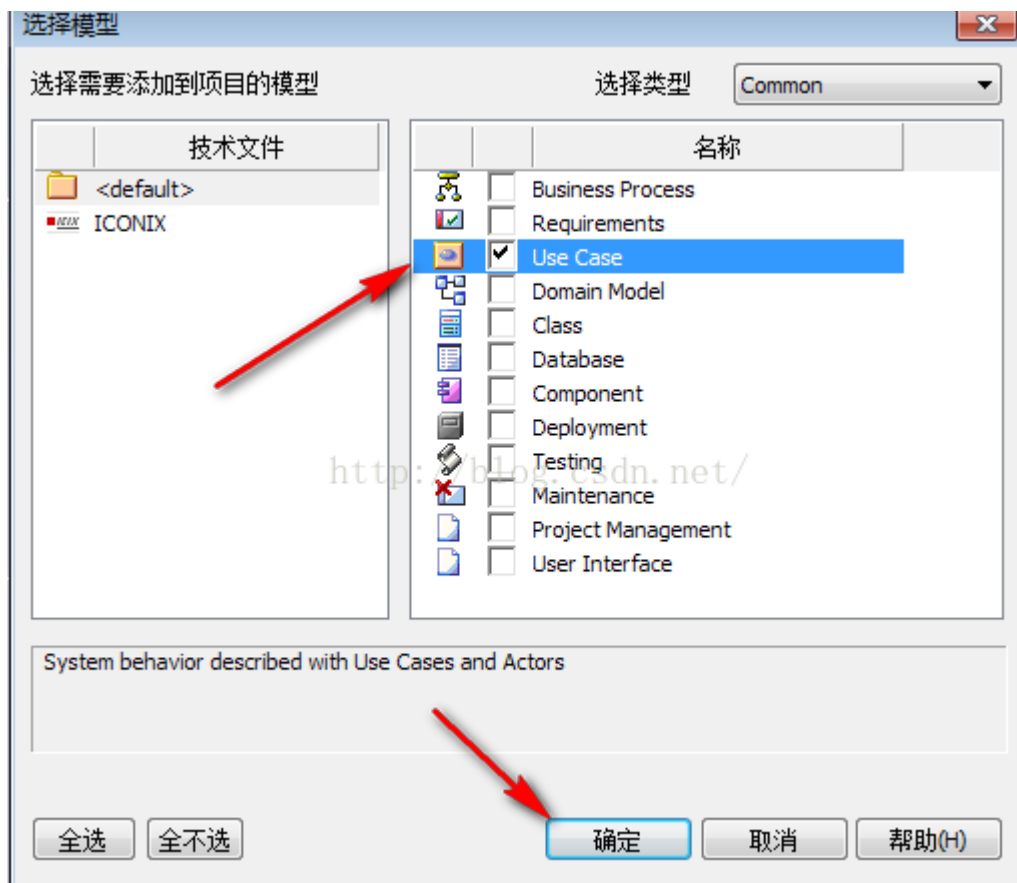


### 3.1.2 用EA画用例图（老师演示例子）

1、选择文件，新建项目



2、选择要保存的目录点击保存，在保存后会跳出选择模型的对话框，这里我选择Use Case，如果这里不选择模型，也可以在建立包的时候选择：

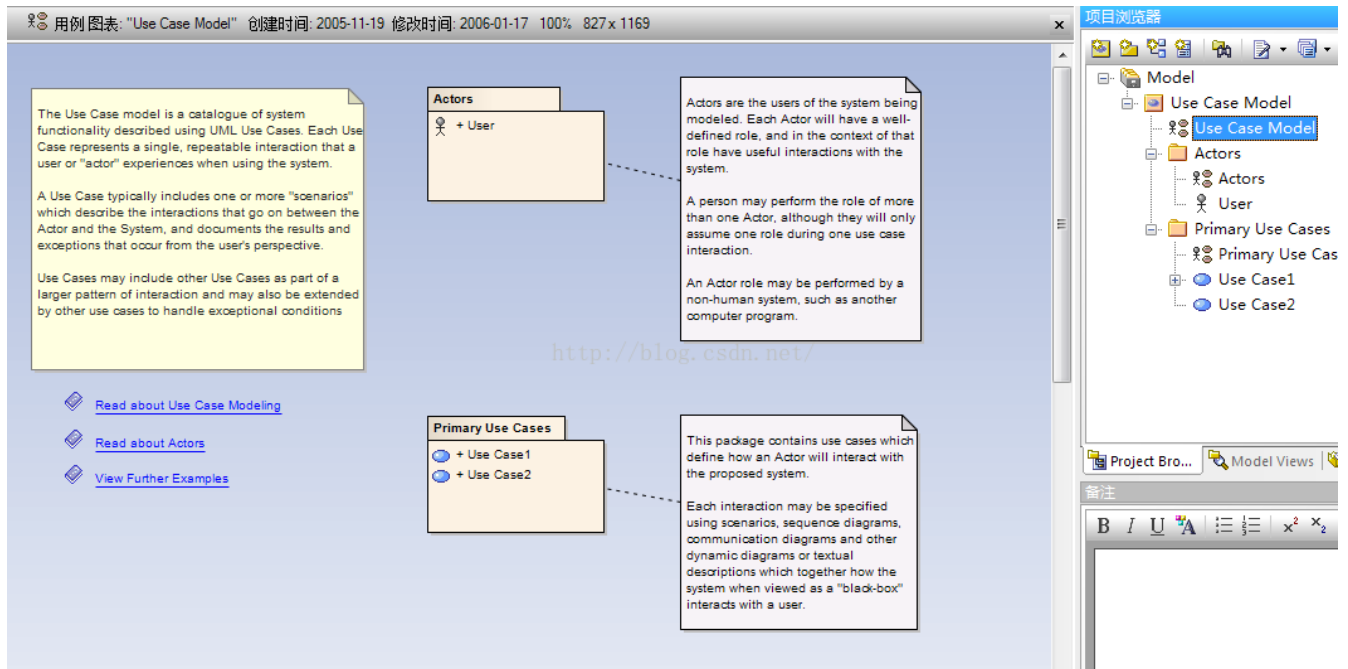


### 3、点确定创建Use Case模型



这里有默认建立模型包及相应的示范用例，这里对示范用例及用例图一般建包原则说明一下

Use Case Model图一般不直接画用例图，一般用来为各包的功能及世中有的用例做说明，如下图：

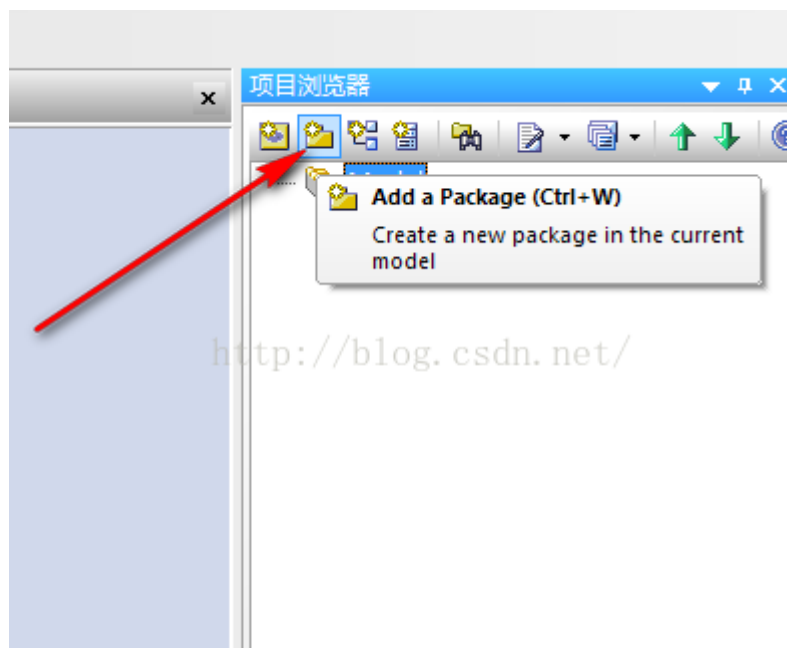


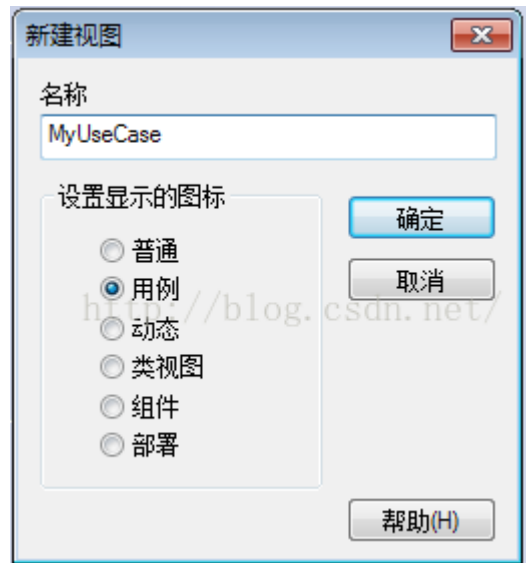
Actors包一般用来放所有的参与者，并为各各参与者，在一般的用例图中参与都一般不多，所以可能不用再分包。

Primary Use Cases包一般用来存放所有的用例，可根据需求分包存放。

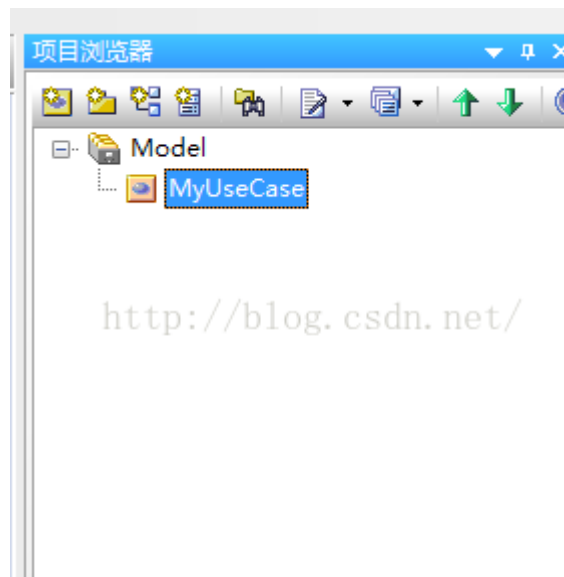
下面使用手动方式创建一个Use Case项目

- 1、新建项目（直接按上面的新建，保存）
- 2、在选择模版的时候点取消，在右侧项目管理器中创建包

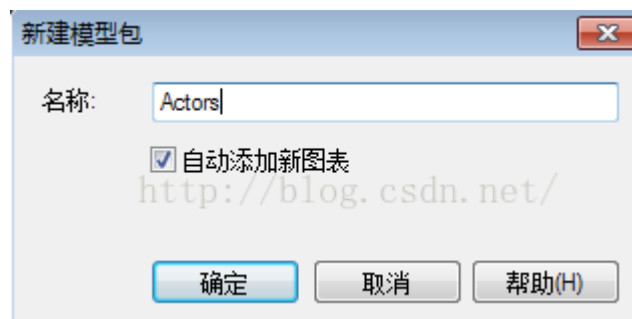




输入名称并选择用例图标按确定

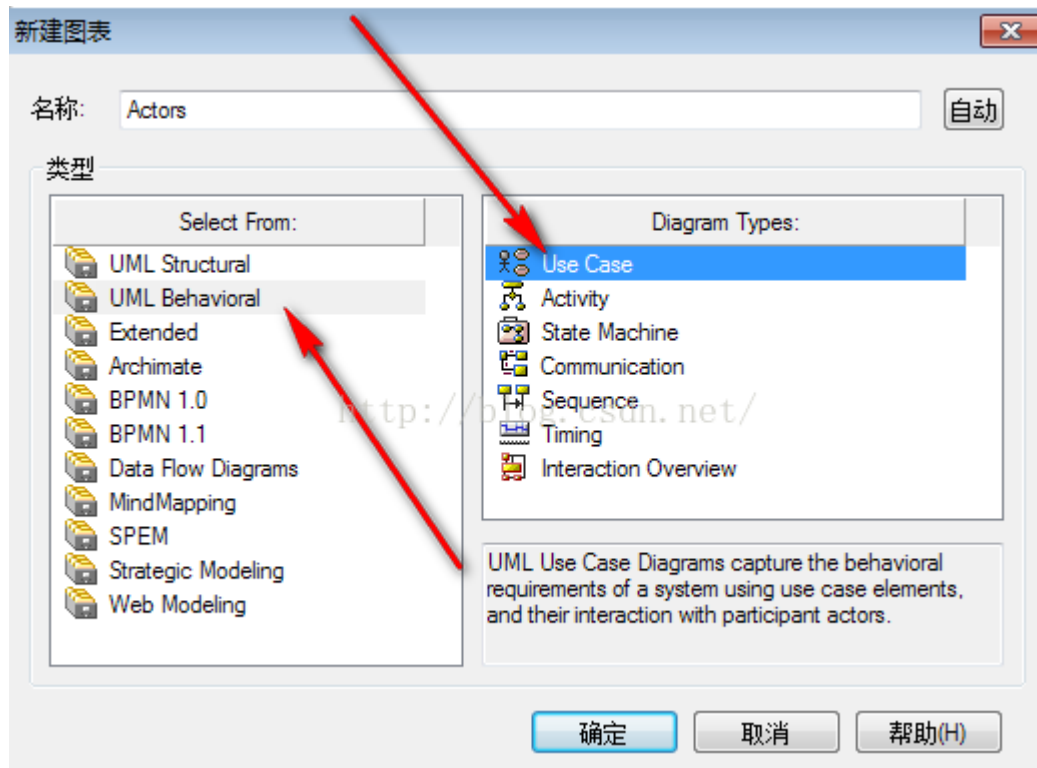


点击Model的+号，并选中刚建的包，还是点击刚才的建包按钮，会跳出新建模型包对话框,输入包名

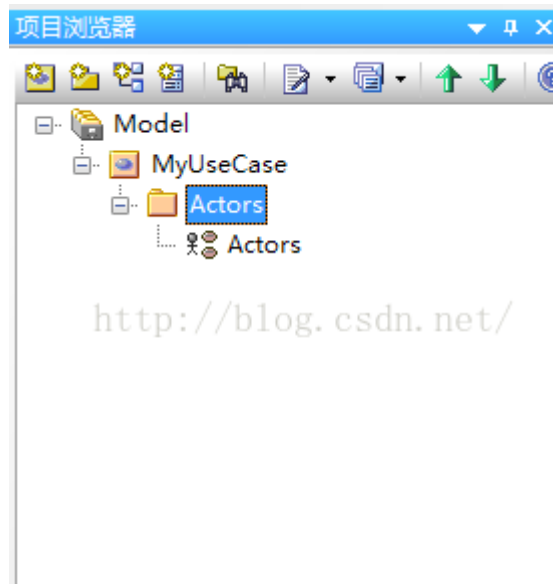


在确定后跳出新建图标窗口，选择相应的模型

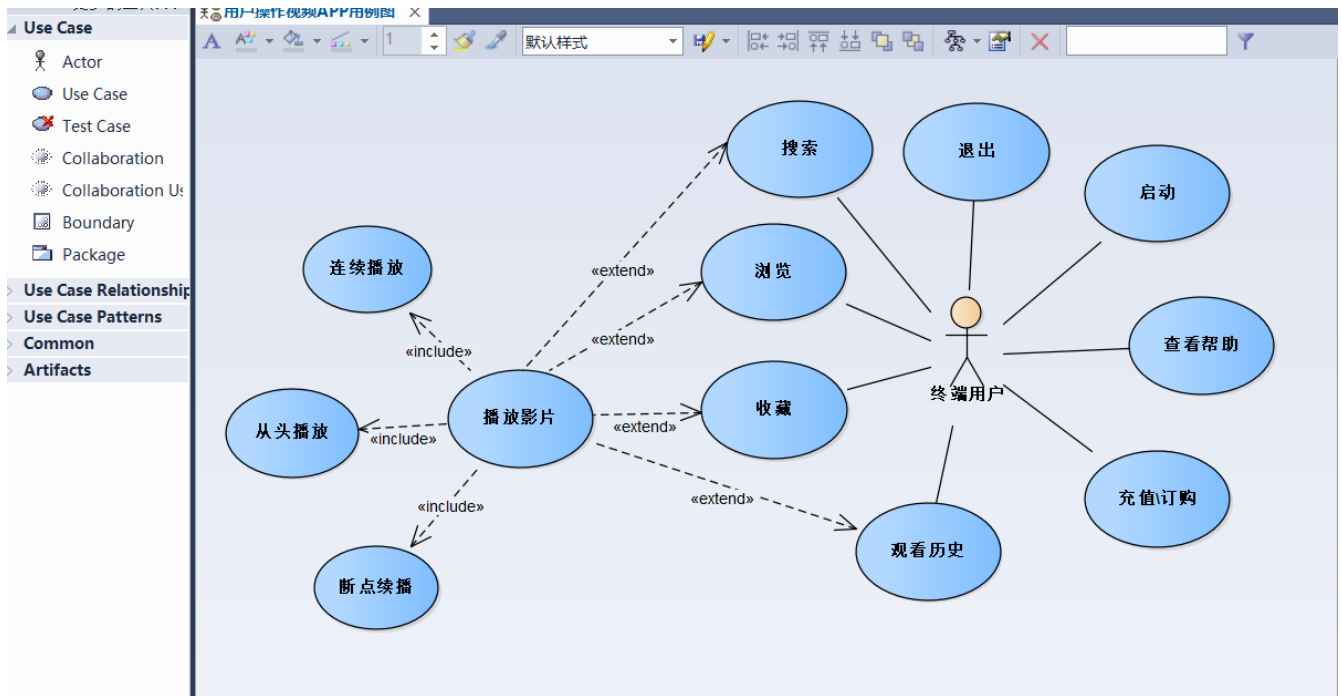




确定后就可看到项目浏览器里新增了一个包和一张图表



4、下图是用户操作某视频应用APP的相关操作（老师用EA演示绘制流程）



### 3.1.3 学生做题

试画出学生成绩管理系统用例图。

参与者角色：教师、学生

用例有：登录、录入成绩、修改成绩、查询成绩

## 3.2 UML组件图

### 3.2.1 怎么画组件图

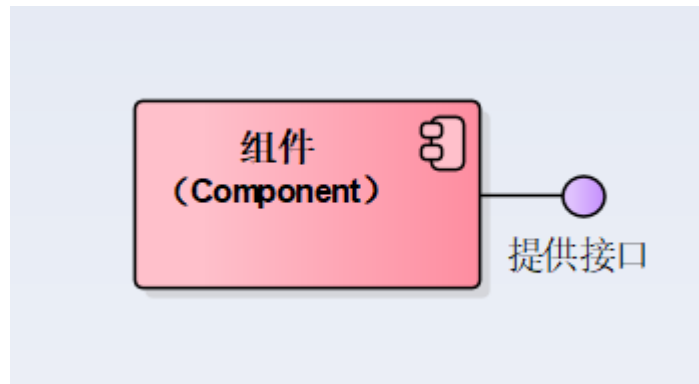
组件图又称为构件图(Component Diagram)。组件图中通常包括组件、接口，以及各种关系。组件图显示组件以及它们之间的依赖关系，它可以用来显示程序代码如何分解成模块或组件。

组件图中主要包含三种元素，即组件、接口和关系。

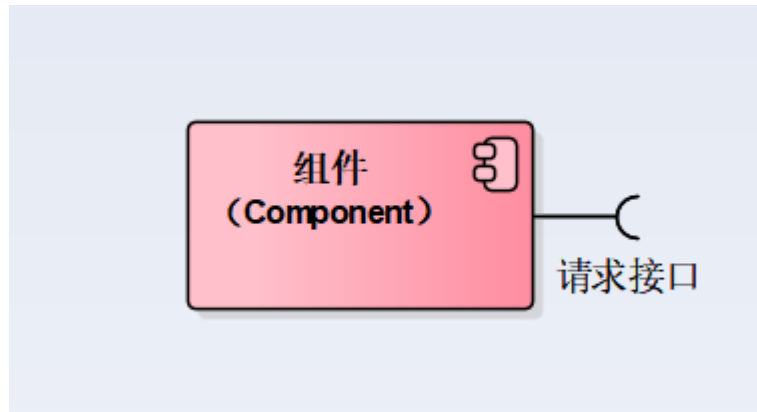
**组件：**系统功能的可重用部件。组件通过接口提供并使用行为，并且可以使用其他组件，如下图所示。



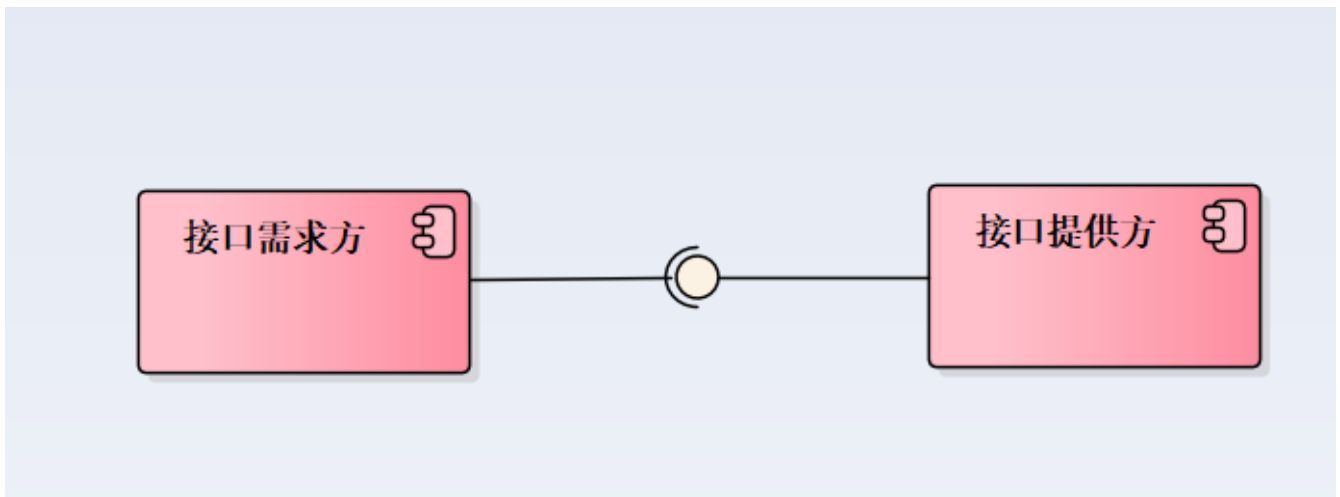
**提供的接口：**表示某个组件实现的且其他组件或外部系统可以使用的一组消息或调用。，如下图所示。



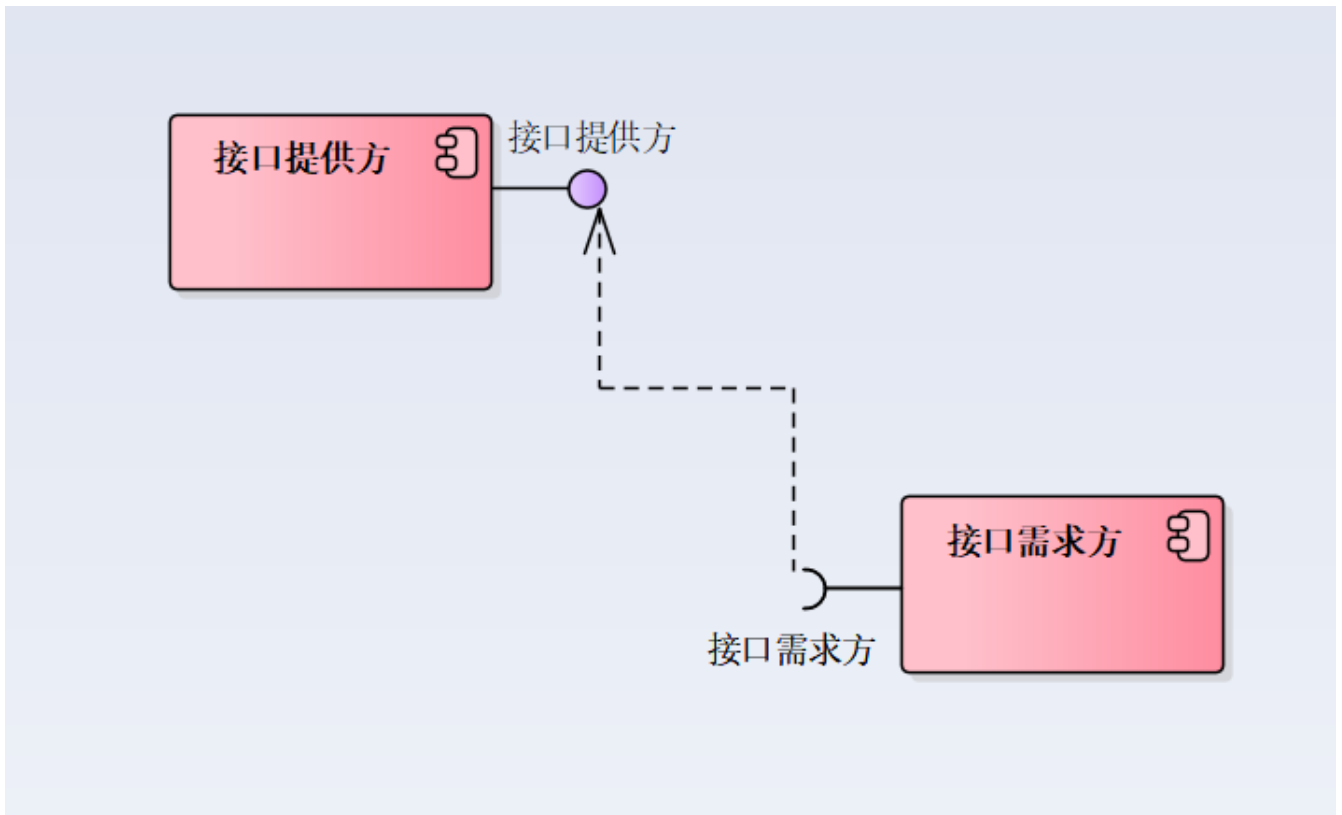
**所需的接口：**表示组件发送到其他组件或外部系统的一组消息或调用，如下图所示。



**依赖关系：**可用于指示一个组件上的所需的接口可以被另一个组件上的提供的接口满足，如下图所示。



如果两个图不在一条水平线上，也可以使用下面的方式来表示：

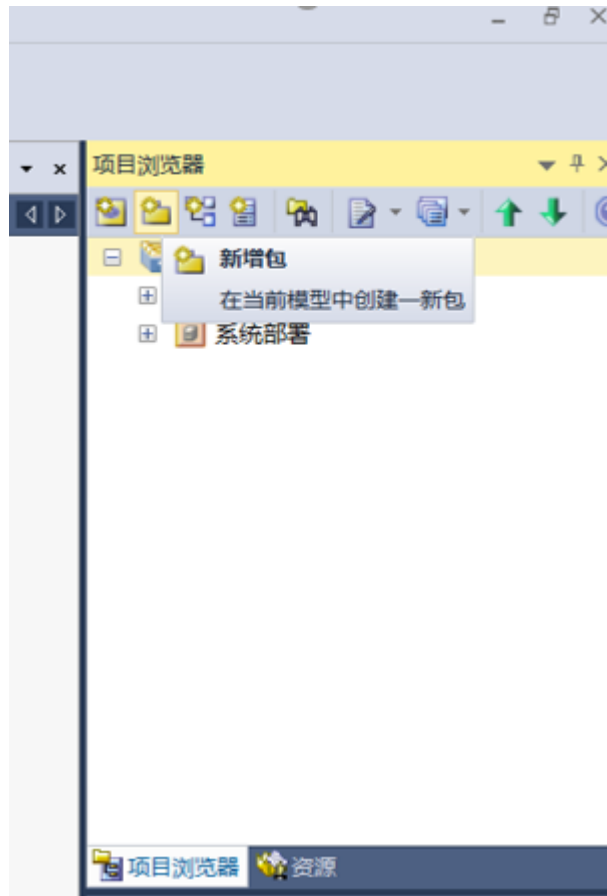


组件图通过这些元素描述了系统的各个组件及之间的依赖关系，还有组件的接口及调用关系。

组件图在面向对象设计过程中起着非常重要的作用：它明确了系统设计，降低了沟通成本，而且按照面向对象方法进行设计的系统和子系统通常保证了低耦合度，提高了可重用性。

### 3.2.2 用EA画时序图（老师演示例子）

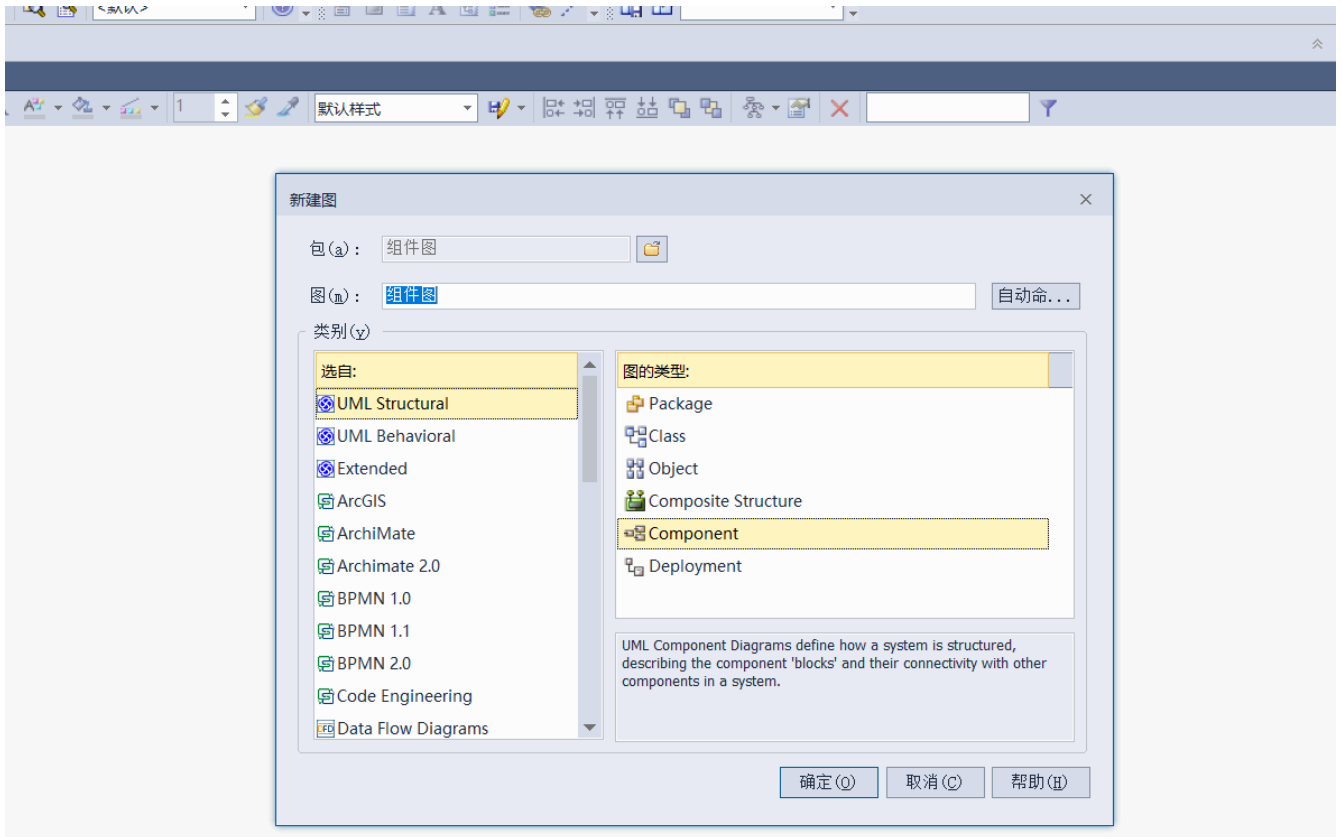
1. 在已建好的项目上侧项目管理器中创建包，如下图所示：



2. 在包上单击右键选择菜单中的“添加图...”,如下图所示:

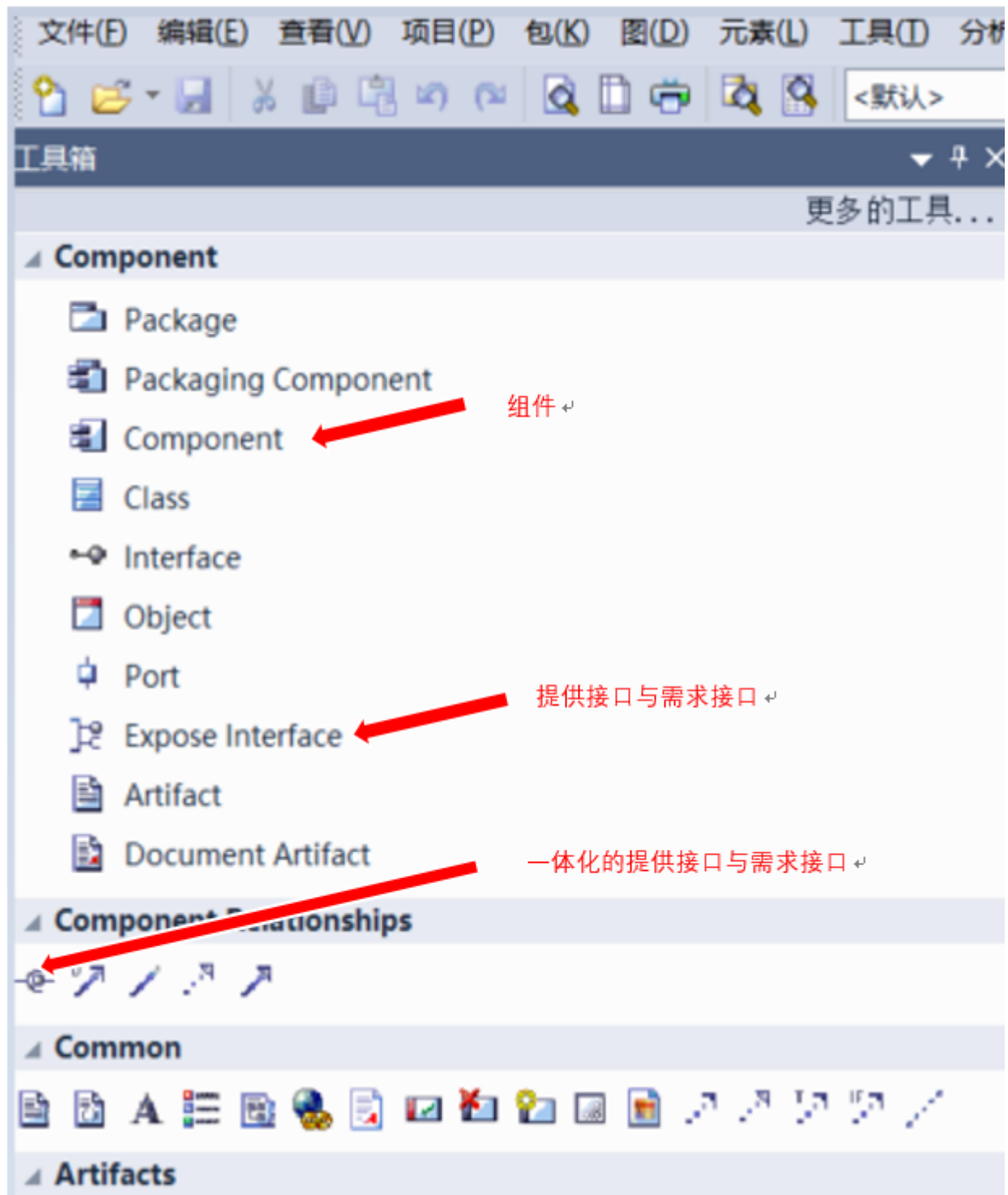


3. 输入图的名称，选择图的类型“Component”，如下图所示：



4.组件图绘制最常用的三个图如下图红色键头所制，从右边工具栏拖出来绘制。





##### 5、网上商城组件图演示（老师用EA演示绘制流程）

案例（分析一个已经存在的系统）

画出下列描述的网上商城组件图：购物车、订单、库存、支付管理组件，使用组件图进行完善。

识别组件：购物车、订单、库存、支付管理

识别组件之间的关系通过一个现实的例子。

在购买一件商品时，我们首先是浏览商品，了解商品详情。在商品详细页面上，我们可以看到一个“加入购物车”



包邮正版/UML和模式应用-(原书第3版)

价格 **¥54.20**  
北京 至 广州 快递: 0.00 平邮: 0.00

0 月销量 0 累计评价  
数量  件 库存5件

立刻购买 **加入购物车**

组件: 购物车  
提供: 加入购物车

服务承诺 按时发货 极速退款 退货保障卡 七天无理由退换 支付方式

组件: 购物车  
所需: 商品库存

已选商品(不含运费): ¥49.50 **结算**

全选	商品	商品详情	单价	数量	小计	操作
<input checked="" type="checkbox"/>	《UML和模式应用(原书第3版)》_拉曼_机械工业出版社		49.50	<input type="text" value="1"/>	49.50	移入收藏夹 删除

组件: 库存  
提供: 商品库存

已选商品 1 件 共 合计(不含运费): ¥49.50 **结算**

组件: 购物车  
所需: 结算

http://blog.csdn.net/gstrong298

确认订单信息

店铺宝贝	单价(元)	数量	优惠方式(元)	小计(元)
《UML和模式应用(原书第3版)》_拉曼_机械工业出版社	49.50	1	无优惠	49.50

结卖家留言:

组件: 支付  
提供: 接收订单

店铺合计(含运费): ¥49.50

使用天猫积分 -0.00

共支付: 支付宝 微信支付 信用卡分期付款 使用天猫点券卡

可获天猫积分: 24点 实付款: **¥49.50**

组件: 订单  
所需: 提交订单

**提交订单**

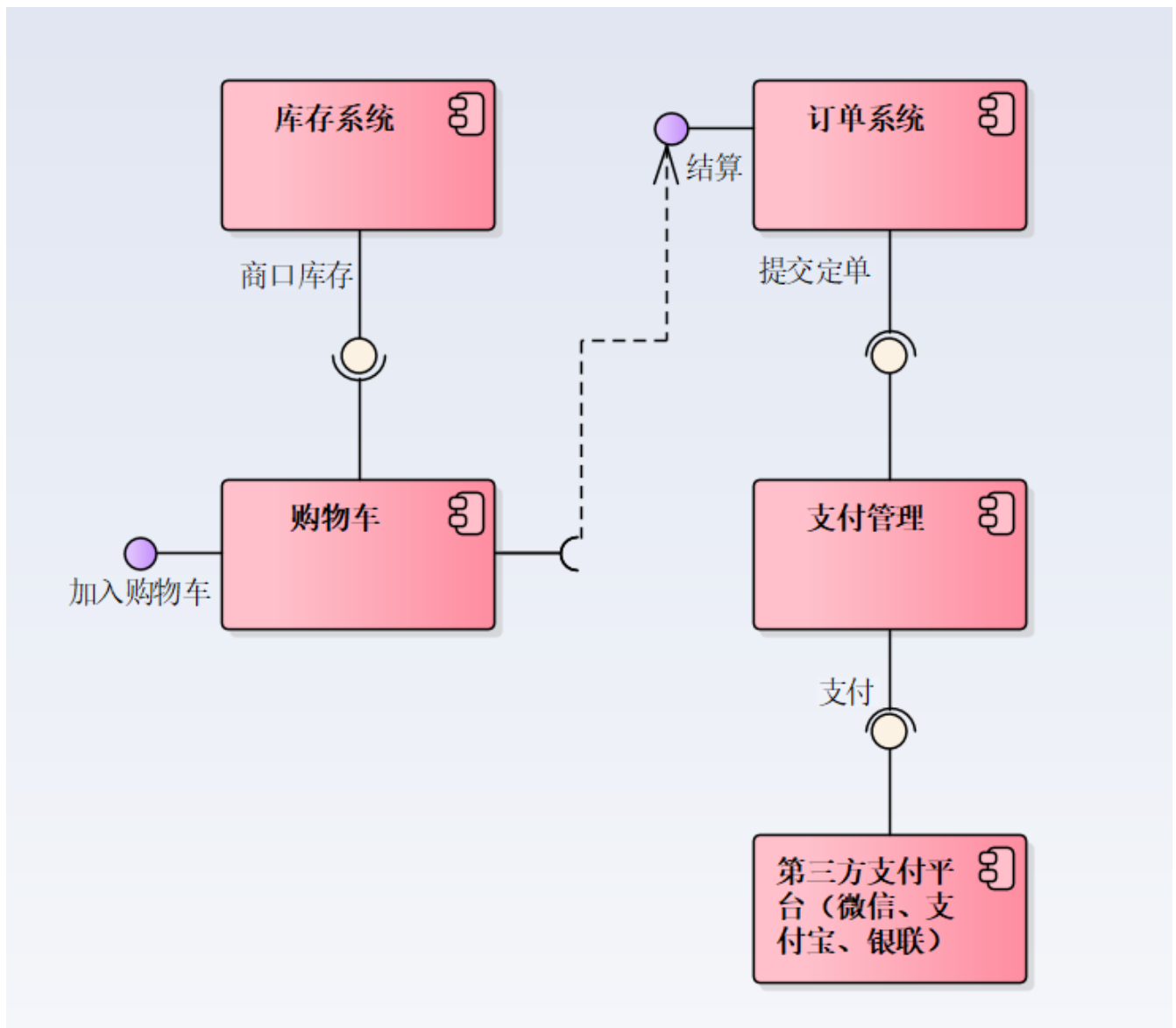
http://blog.csdn.net/gstrong298

The screenshot shows the Alipay 'My Payment Counter' (我的收银台) interface. At the top, it displays the Alipay logo and a QR code. Below the header, a transaction summary is shown: '您正在使用支付宝担保交易 [?]', '天猫Tmall -- [正版]UML和模式应用(原书第3版) \_拉曼\_机械工业出版社', and the amount '49.50 元' with a '刷新' (Refresh) button. A '订单详情' (Order Details) link is also present. The main section shows the '余额宝' (Yu'e Bao) balance and options to '立即转入' (Transfer Immediately) or '立即转出' (Transfer Out Immediately). Below this, there are input fields for '支付宝账户' (Alipay Account) and '可支付余额' (Payable Balance), along with a '使用支付宝购物卡' (Use Alipay Shopping Card) option. A list of payment methods is shown, with '中国工商银行' (Industrial Bank of China) selected and highlighted in orange, showing a balance of '49.50 元'. Other options include '中国建设银行' (China Construction Bank) and '银行卡' (Bank Card). A '下一步' (Next Step) button is at the bottom left.

<http://blog.csdn.net/gstrong298>

各模块提供与需求分析表:

组件	provided	required
购物车	加入购物车	结算
库存	商品库存	\
订单	结算	提交订单
支付管理	接收订单	支付



### 3.2.3 学生做题

某短信公司的短信平台提供对外发短信的接口给客户，短信平台收到短信后经过一些处理后把短信发给运营商，运营商把信息发送到用户手机，然后会返回短信发送状态给短信平台，短信平台收到状态报告后会吧短信推给客户，现在假设有三个组件（客户系统、短信平台、运营商系统）请画出三者组件图和接口调用关系。

## 3.3 UML时序图

### 3.3.1 怎么画时序图

时序图简介 (Brief introduction)

时序图，通过业务处理的参与者的顺序协作来展示软件的不同组件如何协作来完成业务。

时序图 (Sequence Diagram) 是显示对象之间交互的图，这些对象是按时间顺序排列的。顺序图中显示的是参与交互的对象及其对象之间消息交互的顺序。时序图中包括的建模元素主要有：对象 (Actor)、生命线 (Lifeline)、控制焦点 (Focus of control)、消息 (Message) 等等。

#### 1. 角色(Actor)

系统角色，可以是人、及其甚至其他的系统或者子系统。

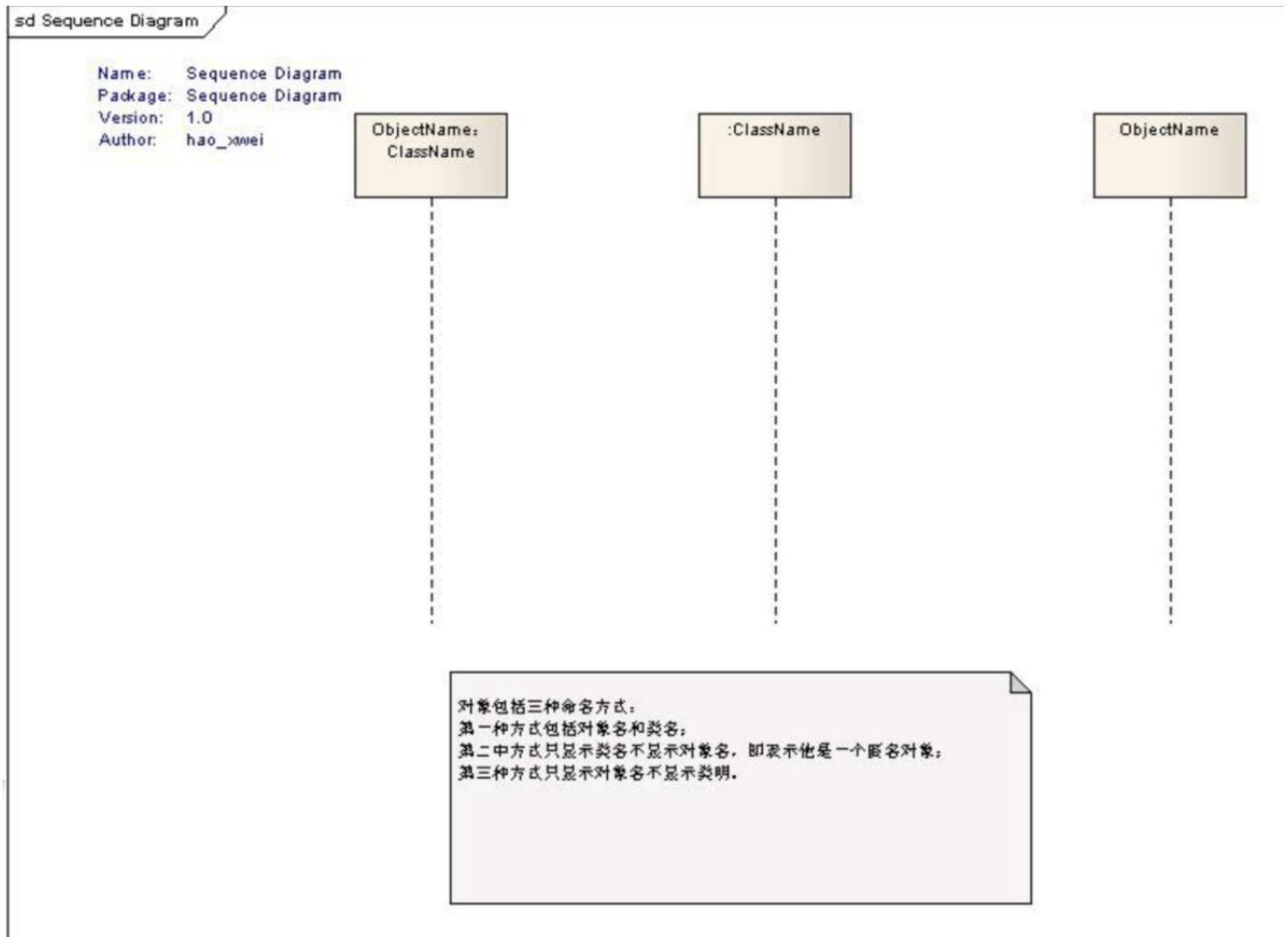
## 2. 对象 (Object)

对象包括三种命名方式：

包括对象名和类名；

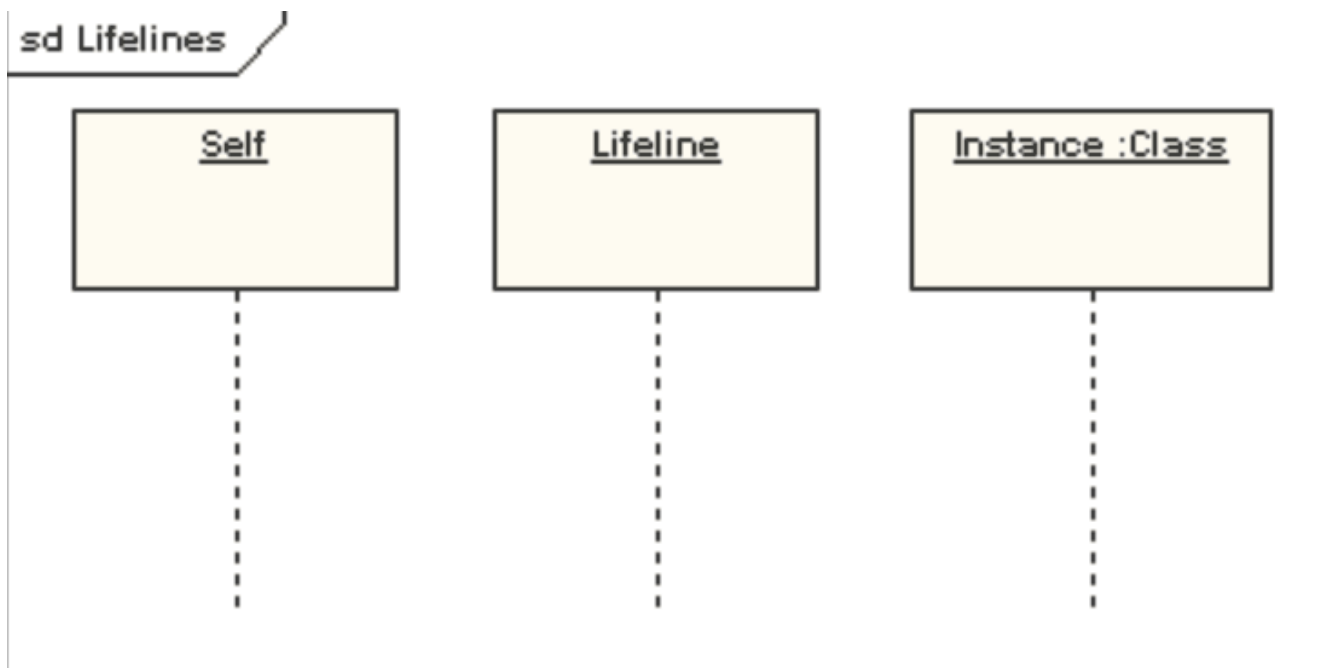
只显示类名不显示对象名，即表示他是一个匿名对象；

只显示对象名不显示类名。



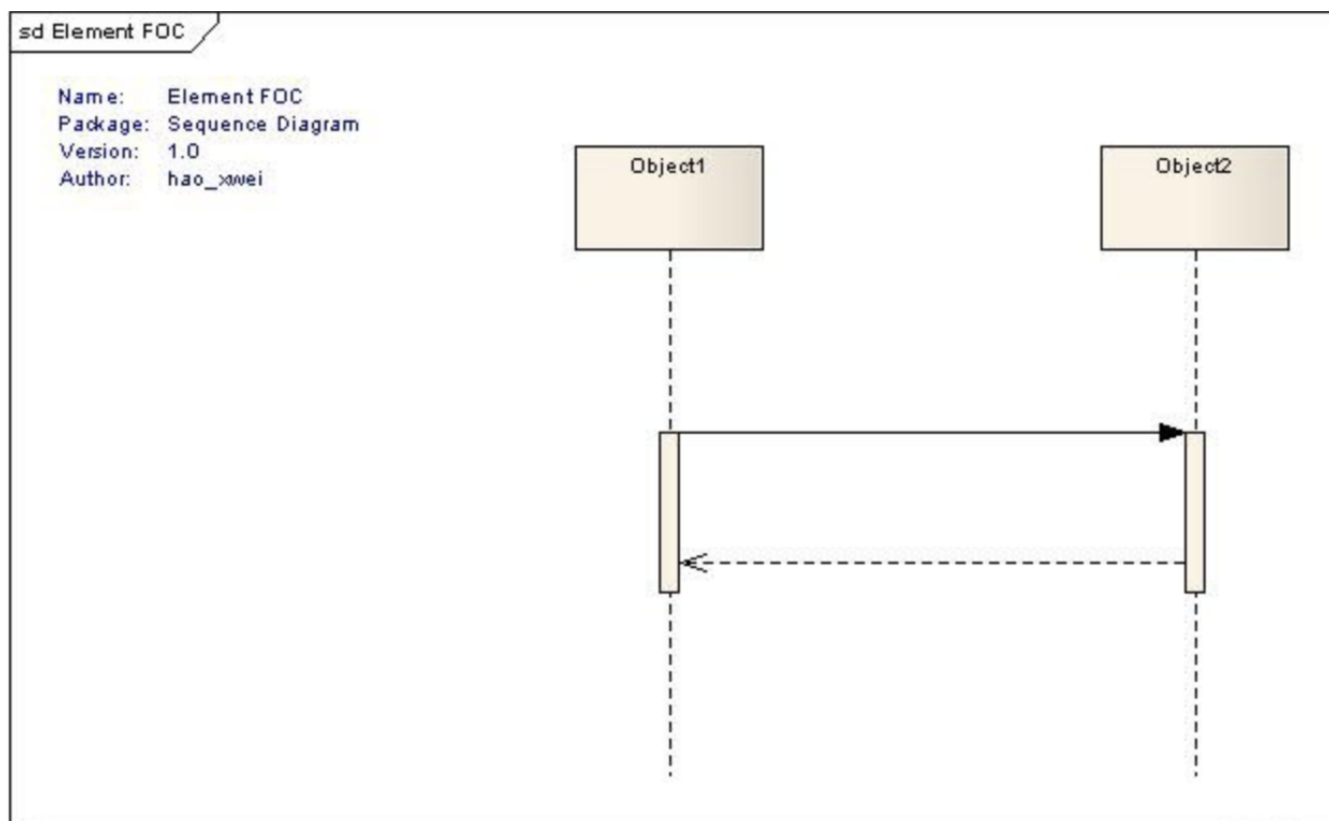
## 3 生命线 (Lifeline)

生命线在顺序图中表示为从对象图标向下延伸的一条虚线，表示对象存在的时间，如下图



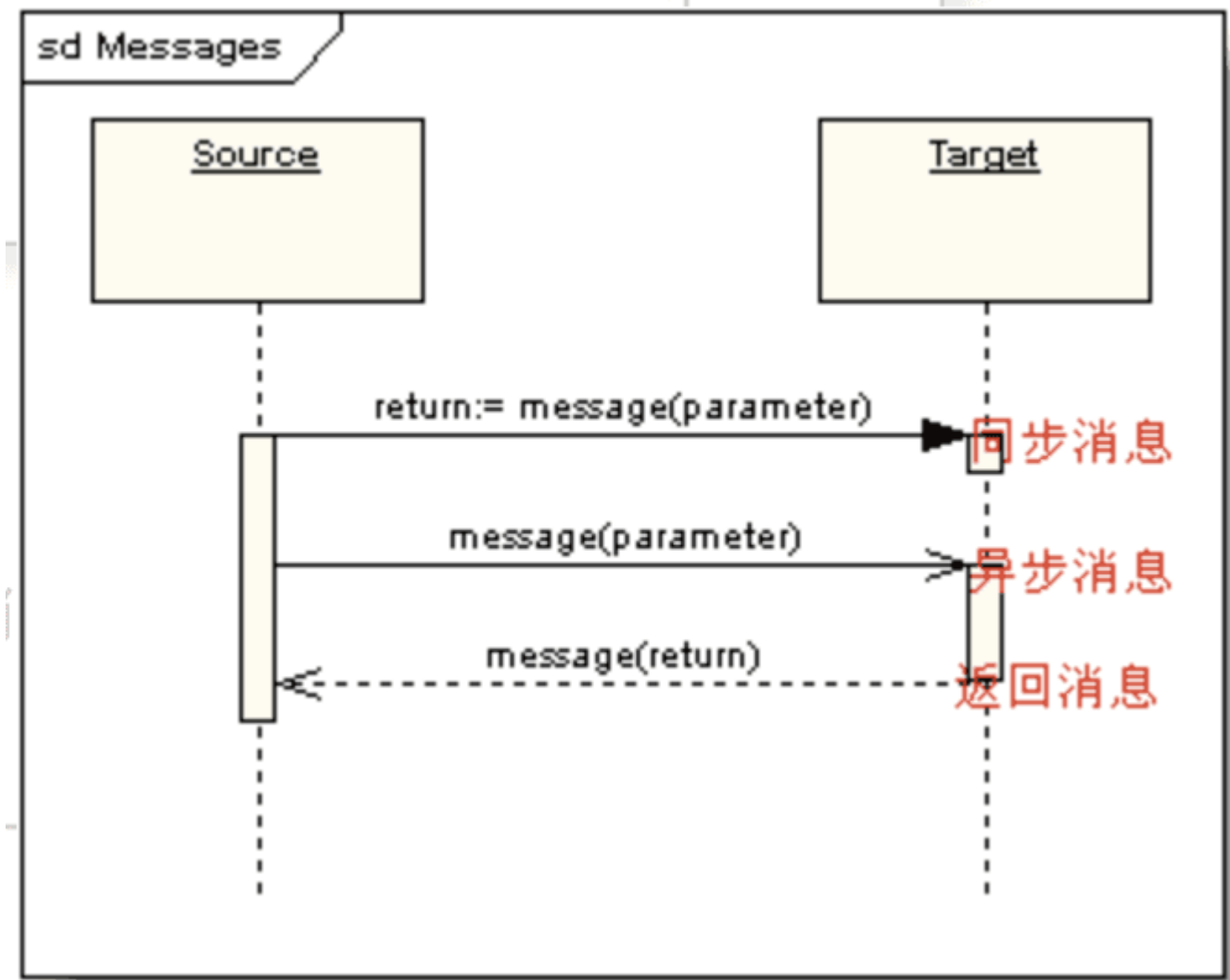
## 2.4 控制焦点 (Focus of Control)

控制焦点是顺序图中表示时间段的符号，在这个时间段内对象将执行相应的操作。用小矩形表示，如下图。



## 2.5 消息 (Message)

消息一般分为 同步消息 (Synchronous Message) , 异步消息 (Asynchronous Message) 和 返回消息 (Return Message) .如下图所示:



#### 同步消息=调用消息 (Synchronous Message) :

消息的发送者把控制传递给消息的接收者，然后停止活动，等待消息的接收者放弃或者返回控制。用来表示同步的意义。

#### 异步消息 (Asynchronous Message) :

消息发送者通过消息把信号传递给消息的接收者，然后继续自己的活动，不等待接受者返回消息或者控制。异步消息的接收者和发送者是并发工作的。

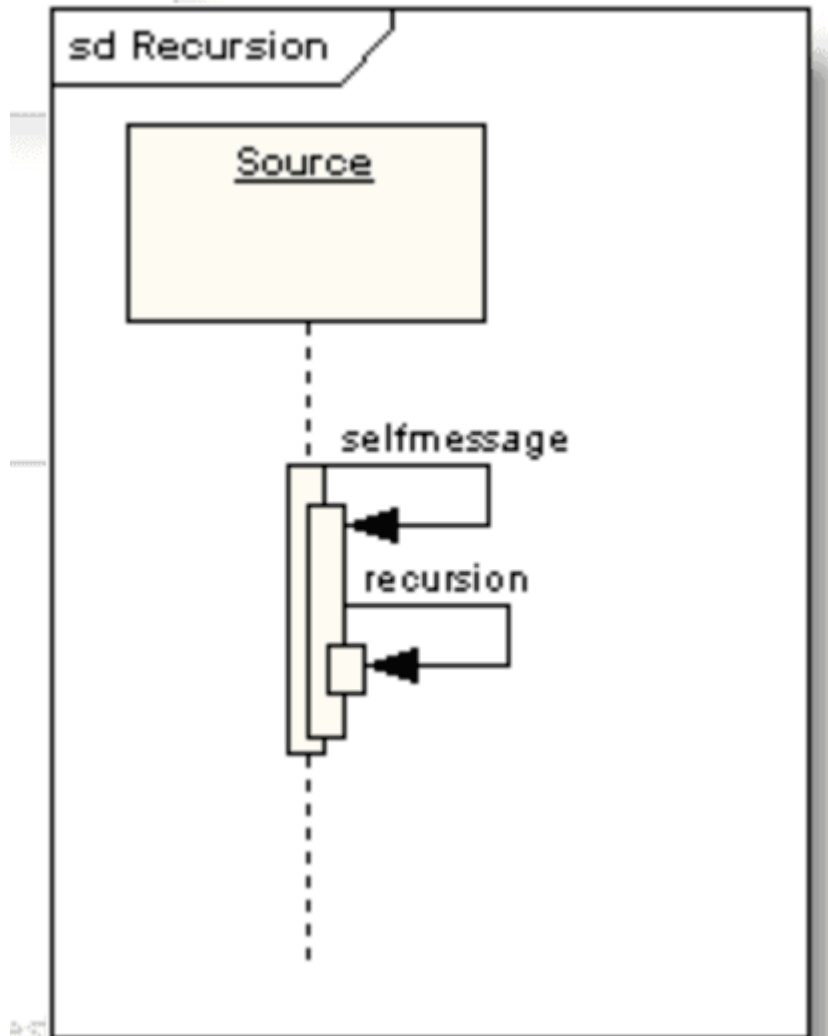
#### 返回消息 (Return Message) :

返回消息表示从过程调用返回

#### 自关联消息 (Self-Message) :

表示方法的自身调用以及一个对象内的一个方法调用另外一个方法。





消息的类型与符号:

简单消息（包括反身消息）



同步消息



异步消息

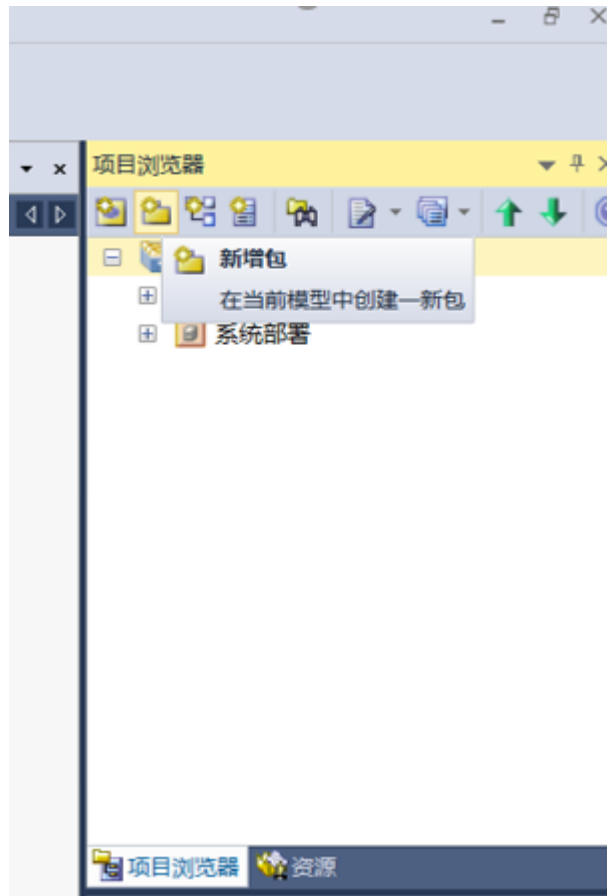


返回消息（可选）



### 3.3.2 用EA画时序图（老师演示例子）

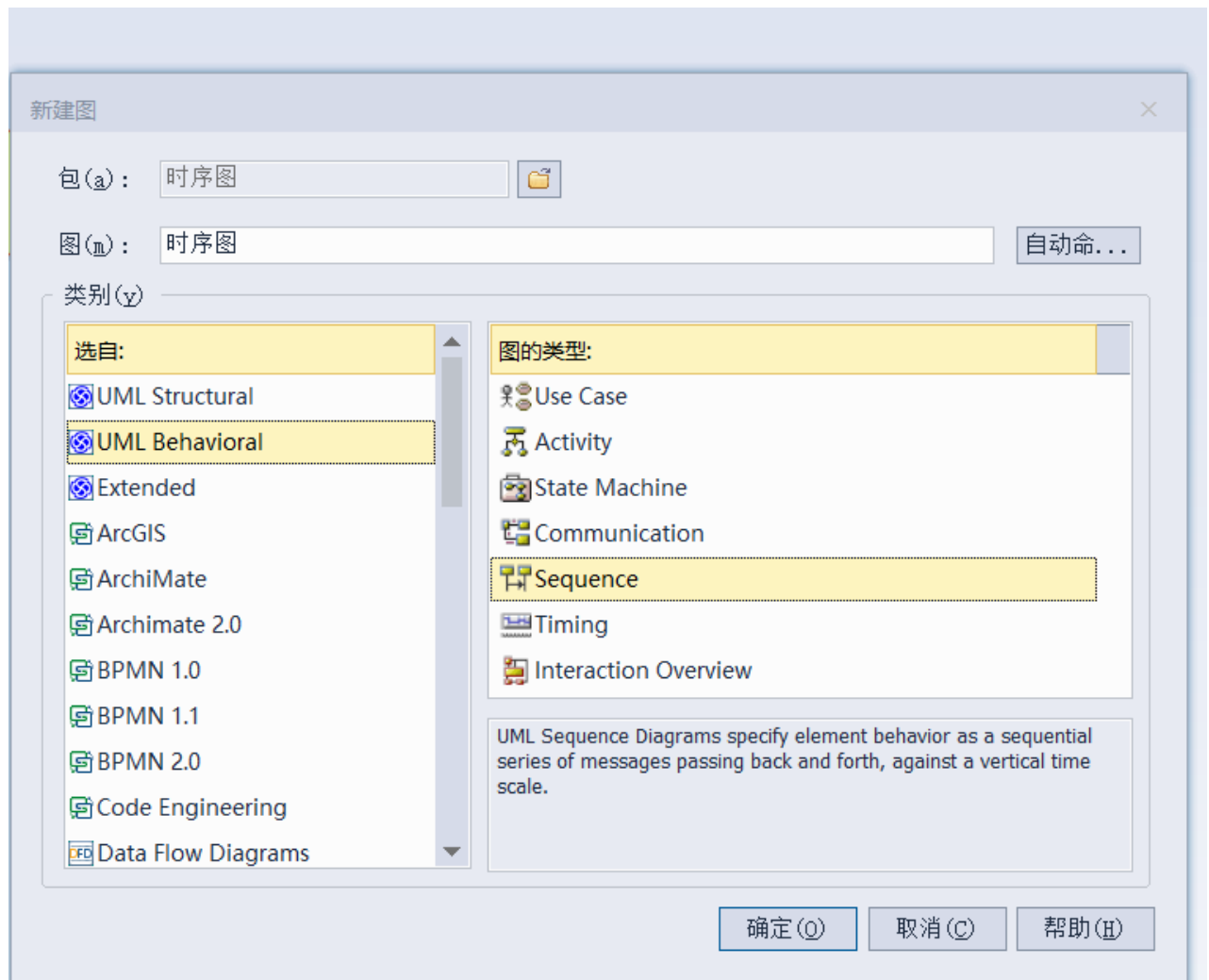
1. 在已建好的项目上侧项目管理器中创建包，如下图所示：



2. 在包上单击右键选择菜单中的“添加图...”,如下图所示:



3. 输入图的名称，选择图的类型“Sequence”，如下图所示：



4. EA中时序图绘制的常用工具见下图红色所示，从右边工具栏拖出来绘制。



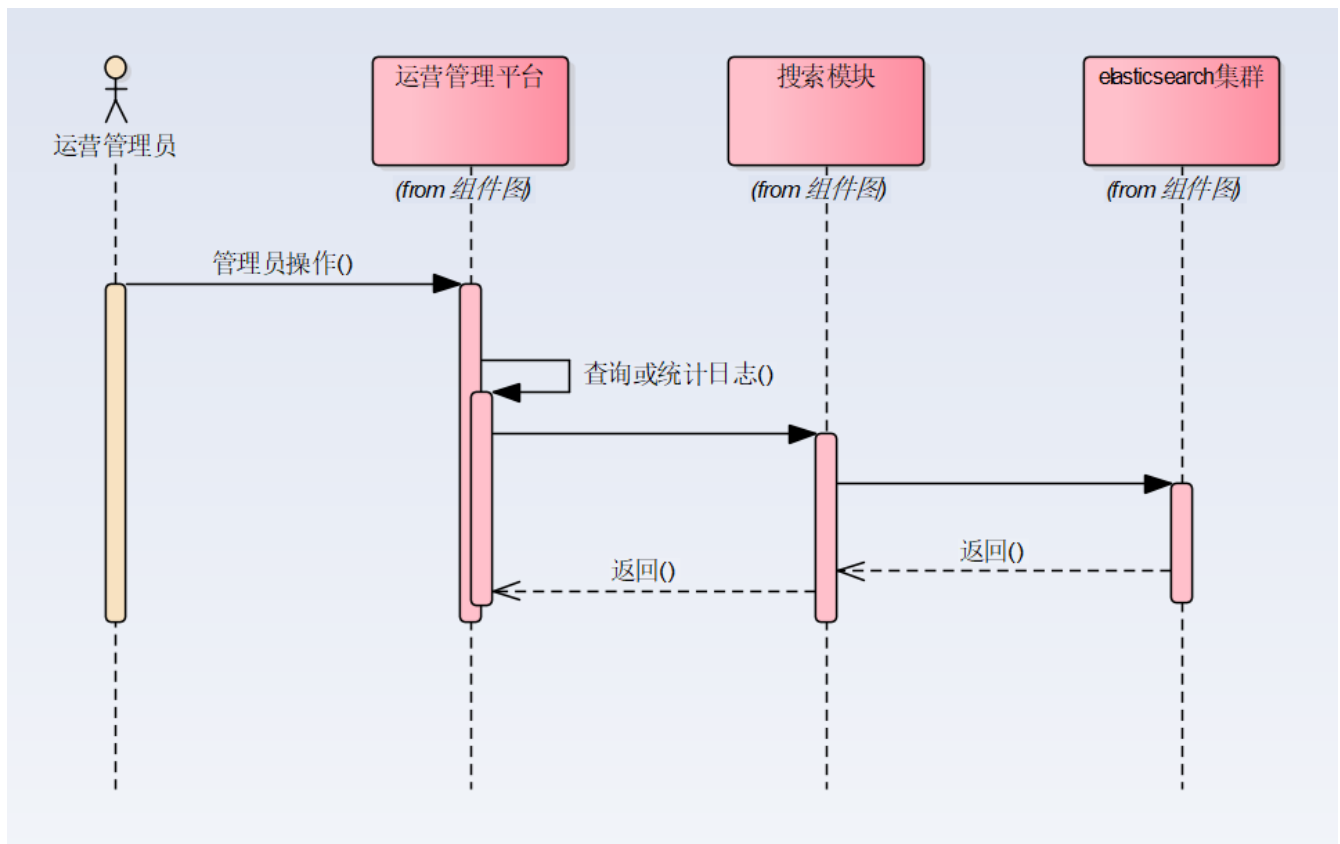
##### 5.运营管理员搜索查询统计日志交互流程（老师用EA演示绘制流程）

运营管理员通过运管管理平台搜索用户请求日志信息，具体交互流程 如下：

第一步，运营管理员在运营管理平台做查询统计；

第二步，运营管理平台调用搜索模块的搜索统计接口

第三步，搜索模块调用ES集群查询返回结果给运营管理平台



### 3.3.3 学生做题

场景：运营管理人员在运营平台修改“应用信息”表的时候同时更新缓存

要求：通用UML的时序图来描述：运营管理员、运营平台、数据库、REDIS缓存四者的交互顺序图

## 3.4 UML活动图（流程图）

### 3.4.1 怎么画活动图

活动图是UML用于对系统的动态行为建模的另一种常用工具，它描述活动的顺序，展现从一个活动到另一个活动的控制流。活动图在本质上是一种流程图。活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。

#### 一、活动图的组成元素 Activity Diagram Element

##### 1、活动状态图 (Activity)

活动状态用于表达状态机中的非原子的运行，其特点如下：

- (1)、活动状态可以分解成其他子活动或者动作状态。
- (2)、活动状态的内部活动可以用另一个活动图来表示。
- (3)、和动作状态不同，活动状态可以有入口动作和出口动作，也可以有内部转移。
- (4)、动作状态是活动状态的一个特例，如果某个活动状态只包括一个动作，那么它就是一个动作状态。

UML中活动状态和动作状态的图标相同，但是活动状态可以在图标中给出入口动作和出口动作等信息。

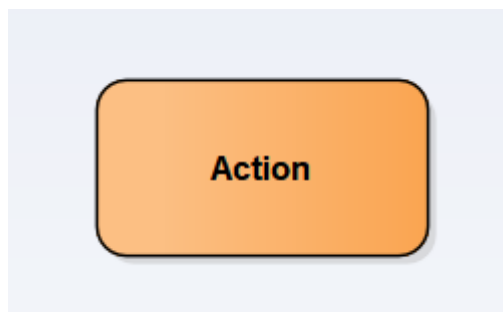


## 2、动作状态 (Actions)

动作状态是指原子的，不可中断的动作，并在此动作完成后通过完成转换转向另一个状态。动作状态有如下特点：

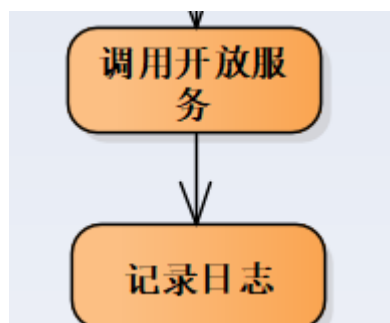
- (1)、动作状态是原子的，它是构造活动图的最小单位。
- (2)、动作状态是不可中断的。
- (3)、动作状态是瞬时的行为。
- (4)、动作状态可以有入转换，入转换既可以是动作流，也可以是对象流。动作状态至少有一条出转换，这条转换以内部的完成为起点，与外部事件无关。
- (5)、动作状态与状态图中的状态不同，它不能有入口动作和出口动作，更不能有内部转移。
- (6)、在一张活动图中，动作状态允许多处出现。

UML中的动作状态图用平滑的圆角矩形表示，如下：



## 3、动作流 (Control Flow)

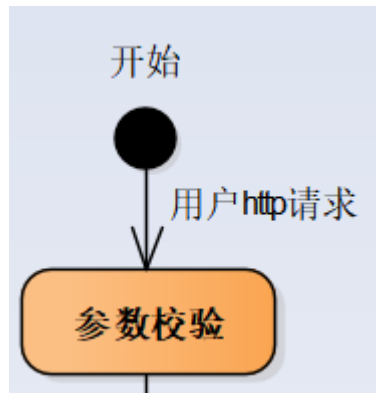
动作之间的转换称之为动作流，活动图的转换用带箭头的直线表示，箭头的方向指向转入的方向。



## 4、开始节点 (Initial Node)

开始节点：表示成实心黑色圆点

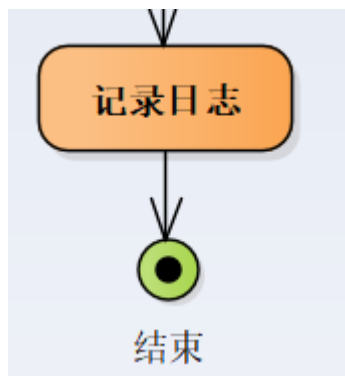




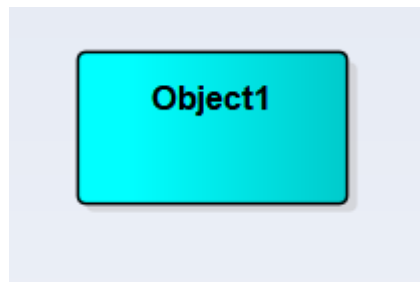
#### 5、终止节点 (Final Node)

分为活动终止节点 (activity final nodes) 和流程终止节点 (flow final nodes)。

活动终止节点表示整个活动的结束



#### 6、对象 (Objects)



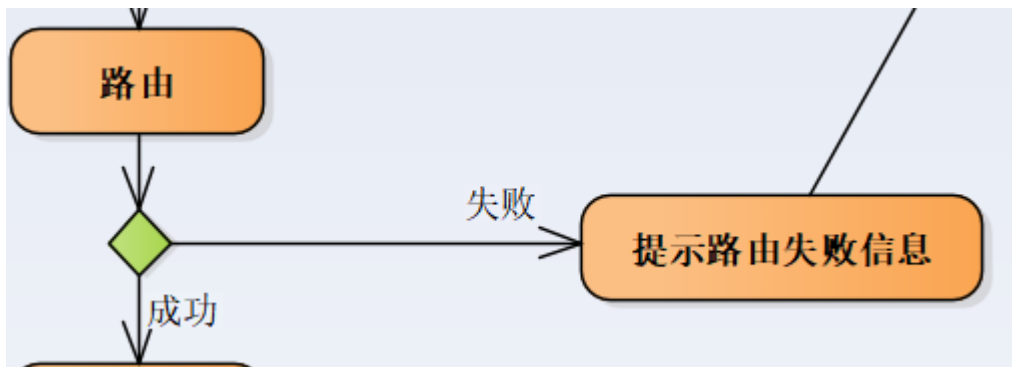
#### 8、数据存储对象 (DataStore)



使用关键字datastore

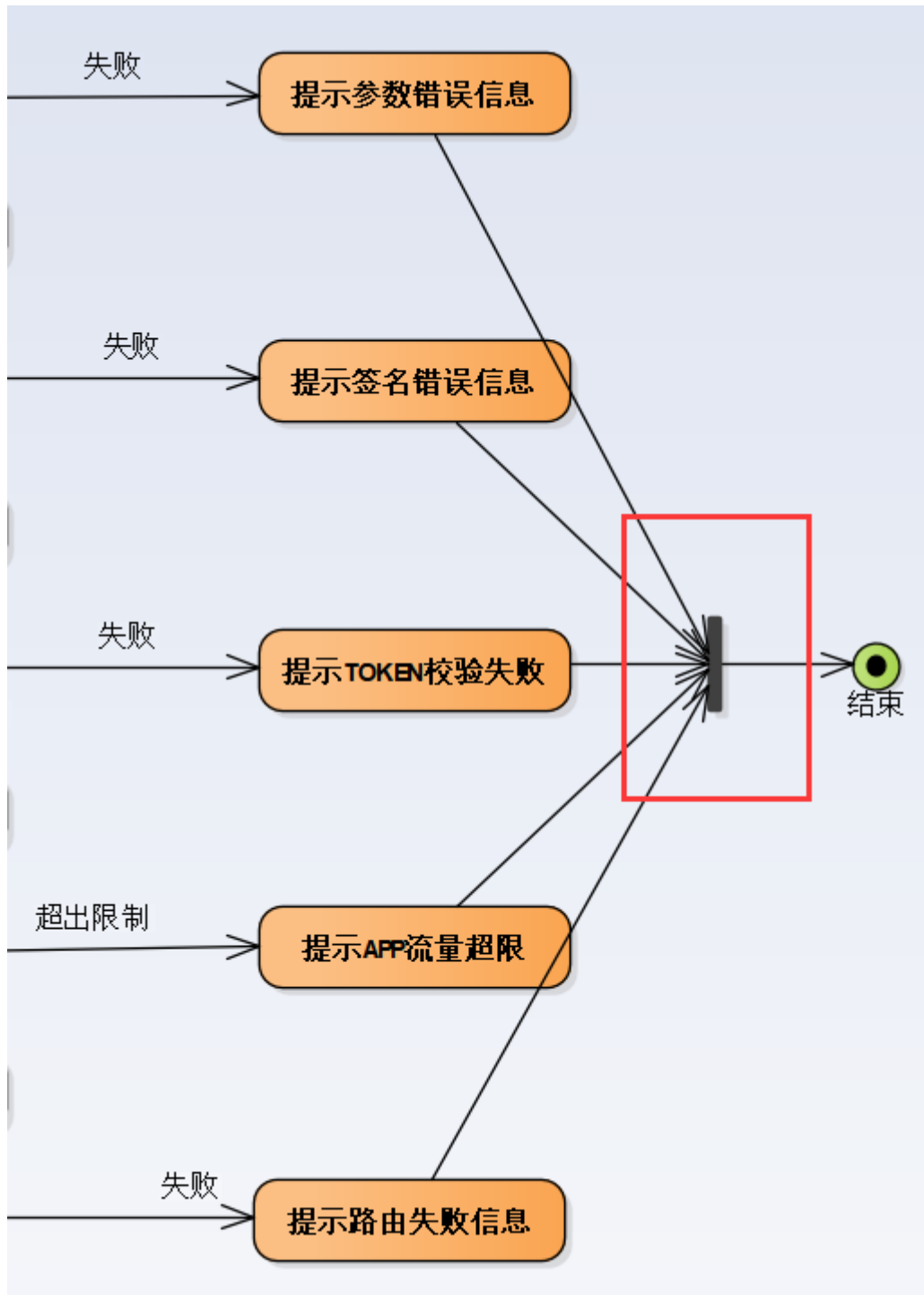
#### 9、分支与合并 (Decision and Merge Nodes)

分支与合并用菱形表示，它有一个进入转换（箭头从外指向分支符号），一个或多个离开转换（箭头从分支符号指向外）。而**每个离开转换上都会有一个监护条件**，用来表示满足什么条件的时候执行该转换。



#### 11、分叉与汇合 (Fork and Join Nodes)

分为水平风向和垂直方向。对象在运行时可能会存在两个或多个并发运行的控制流，为了对并发的控制流建模，UML中引入了分叉与汇合的概念。分叉用于将动作流分为两个或多个并发运行的分支，而汇合则用于同步这些并发分支，以达到共同完成一项事务的目的。

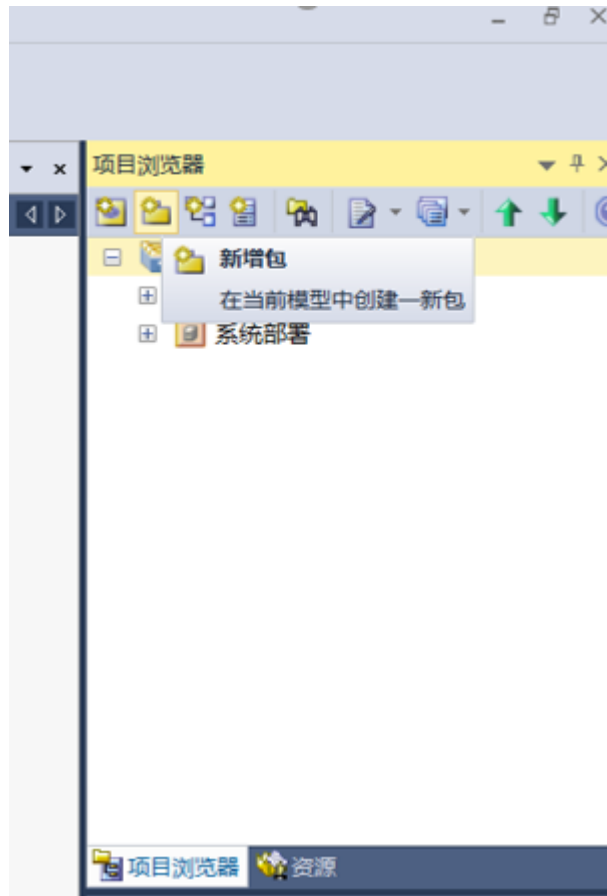


#### 14、泳道 (Partition)

泳道将活动图中的活动划分为若干组，并把每一组指定给负责这组活动的业务组织，即对象。在活动图中，泳道区分了负责活动的对象，它明确地表示了哪些活动是由哪些对象进行的。在包含泳道的活动图中，每个活动只能明确地属于一个泳道。

泳道是用垂直实线绘出，垂直线分隔的区域就是泳道。在泳道的上方可以给出泳道的名字或对象的名字，该对象负责泳道内的全部活动。泳道没有顺序，不同泳道中的活动既可以顺序进行也可以并发进行，动作流和对象流允许穿越分隔线。

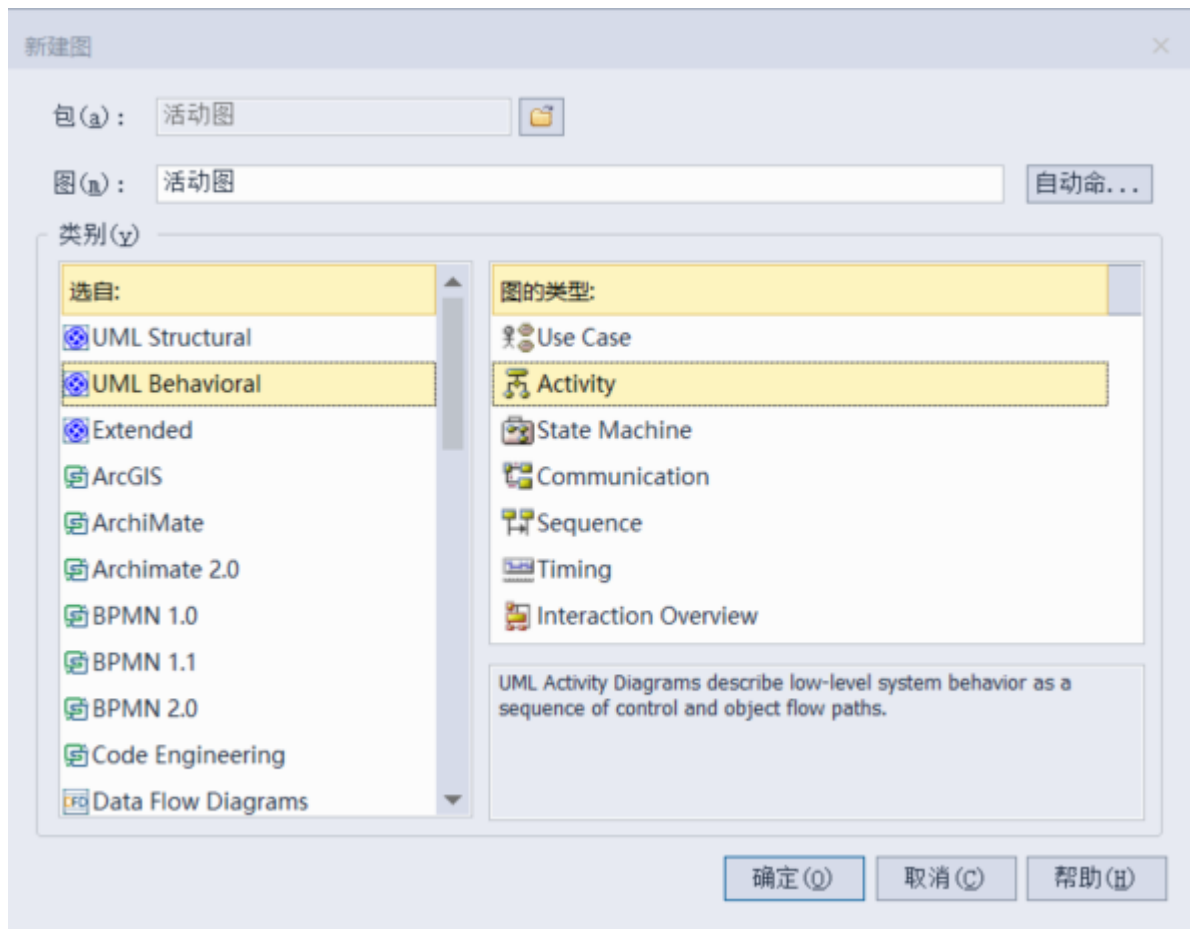
1. 在已建好的项目上侧项目管理器中创建包，如下图所示：



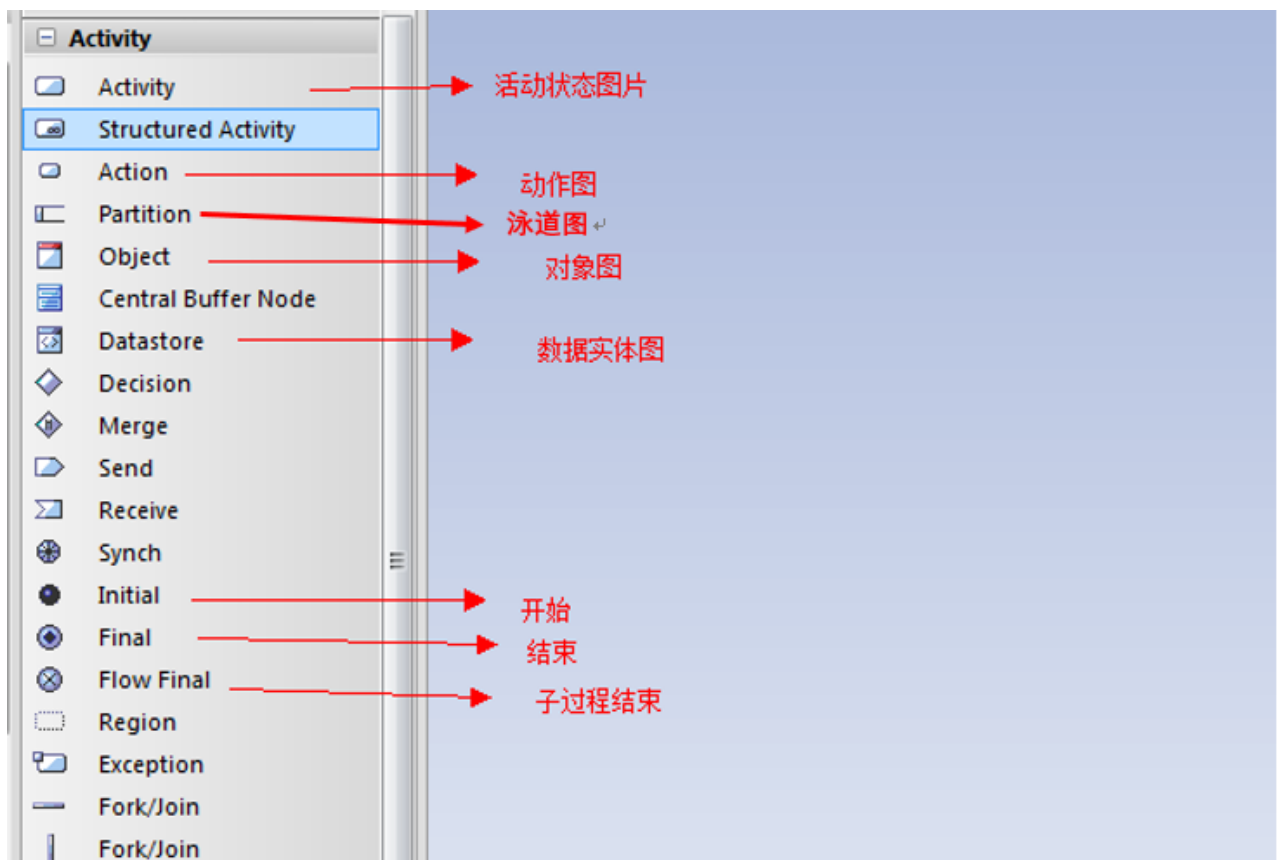
2. 在包上单击右键选择菜单中的“添加图...”,如下图所示:



3. 输入图的名称，选择图的类型“Activity”，如下图所示：

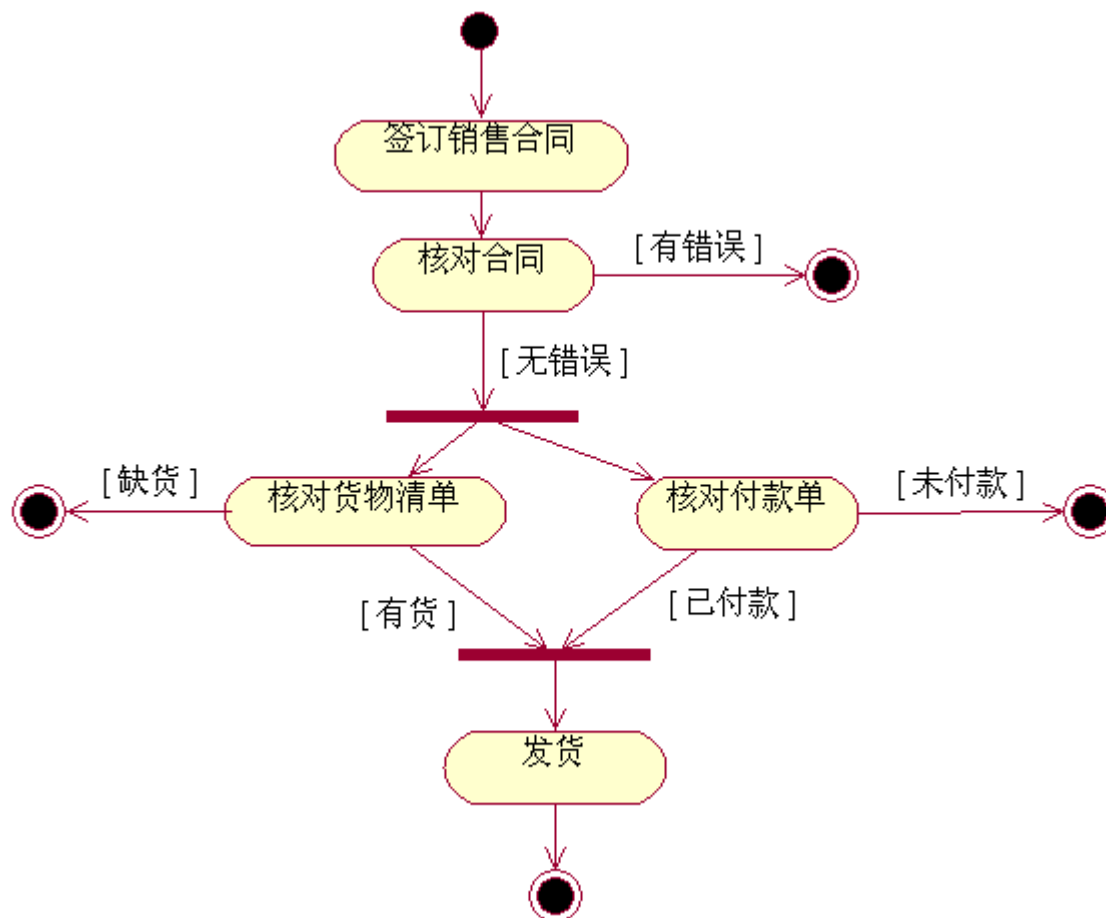


4. EA中时序图绘制的常用工具见下图红色所示，从右边工具栏拖出来绘制。



5. 销售合同从签订到履约的过程（老师用EA演示绘制流程）

销售合同签订后，要进行核对。如果发现错误，则终止履约；如果没有错误，则要核对货物清单确定是否有货，还要核对付款单确定对方是否已经付款，只有这两项都完成，才可以发货。如果无货或对方尚未付款，则终止履约。则活动图如下：



### 3.4.3 学生做题

客户访问一个开放平台的API得通过网关来做校验，校验流程 如下：

1. 首先要做密码校验，如果错误就提示错误信息并退出
2. 接下来做流量限制，如果超量就退出
3. 最后查询后结束流程

请画出流程图。

## 第四章 随堂项目背景介绍

详见“学生实习综合项目\开放平台\开放平台总体介绍说明书.doc”文档。

## 第五章 开放平台设计说明

详见“学生实习综合项目\开放平台\淘宝电商开放云平台(网关部分)总体设计说明书.doc”文档。



## 第六章 学生课后作业

---

1. 反复练习老师演示的和学生课堂中画的几张图
2. 结合学到的UML知识阅读分析开放平台的需求文档和设计文档。

## 第七 面试题

---

- 一、平常工作中都用到了哪些UML图、起什么作用？
- 二、画出你做的开放平台的架构图（此题可以在做完项目后再做）？
- 三、什么是Oauth2协议，它的作用是什么？