UML 模型框架（基于 EA）

作者：黄月 俎涛（火龙果软件工程）

说明：

- 模型来自 Architecture Enterprise Demo
- 框架有关的说明来自 EA 技术资料

关键词：UML，EA，建模，模型框架

目录

UML 模型框架（基于 EA）.....1
1. UML 概览.....2
2.EA 对 UML 建模的支持.....3
2.1 静态结构图.....3
2.1.1 PIM 类图模型示例（Class Diagram）.....3
2.1.2 组件图模型示例（Component Diagram）.....8
2.1.3 包图模型示例（Package Diagram）.....11
2.1.4 复合结构图模型示例（Composite Structure Diagram）.....12
2.1.5 对象图模型示例（Object Diagram）.....15
2.1.6 部署图模型示例（Deployment Diagram）.....16
2.2 动态行为图.....17
2.2.1 用例图模型示例（Use Case Diagram）.....17
2.2.2 活动图模型示例（Activity Diagram）.....19
2.2.3 顺序图模型示例（Sequence Diagram）.....20
2.2.4 通信图模型示例（Communication Diagram）.....21
2.2.5 交互概览图模型示例（Interaction Overview Diagram）.....22
2.2.6 状态图模型示例（StateMachine Diagram）.....23
2.2.7 时间图模型示例（Time Diagram）.....24

1.UML 概览

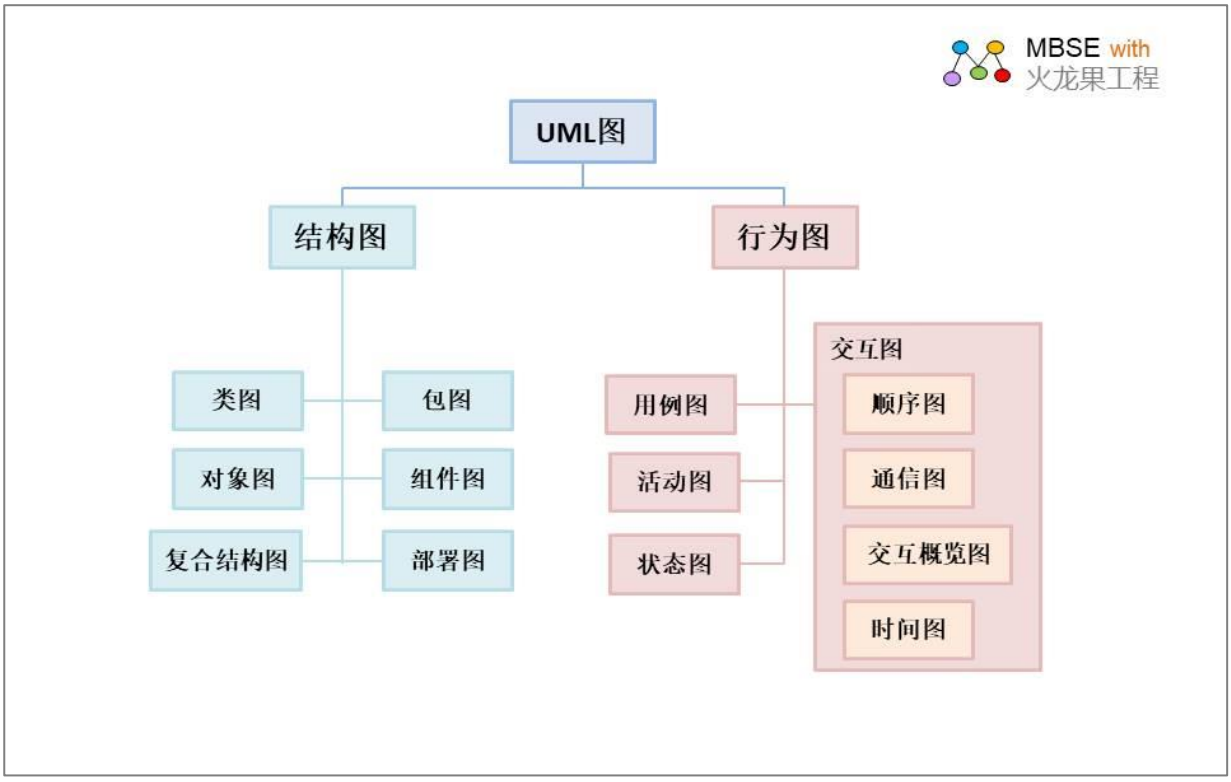
UML（Unified Modeling Language）又称统一建模语言或标准建模语言，是一个支持模型化和软件系统开发的图形化语言，为软件开发的所有阶段提供模型化和可视化支持，包括从需求分析到规格，再到构造和配置。

UML 当前最新规范是 2.5，共有 13 种图，分为两大类：静态结构图和动态行为图。

- 静态结构图
  这类图形通常用来对那些构成模型的“要素”进行建模，诸如：类，对象，接口和物理组件。另外，它们也被用来对元素间的关联和依赖关系进行建模。共有 6 种：类图、对象图、组件图、包图、部署图、复合结构图。

- 动态行为图
  这类图形用来记录在一个模型内部，随时间的变化，模型执行的交互变化和交互瞬间的状态；并跟踪系统在真实环境下如何表现，以及观察系统对一个操作或事件的反应和结果。共有 7 种：用例图、活动图、状态图、顺序图、通信图、交互概览图、时间图，其中后 4 种图又统称为交互图。

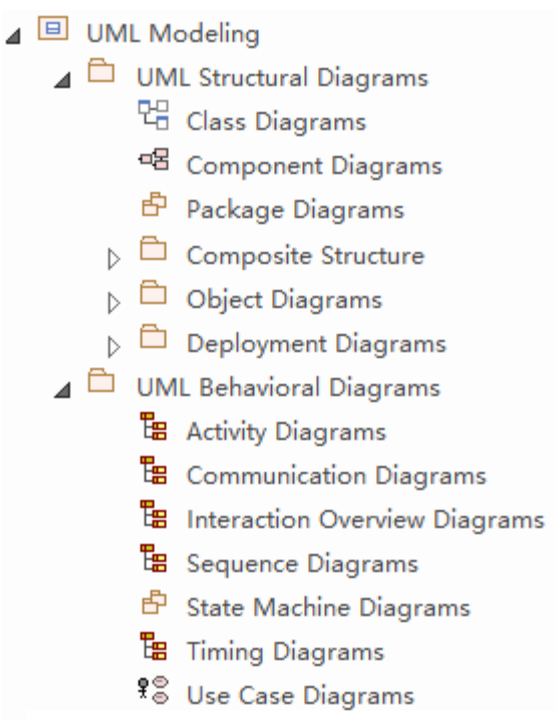
如下图所示：



2.EA 对 UML 建模的支持

Enterprise Architecture 提供了一套完整的 UML 模型框架，并能将模型进行转换。本文通过 EA 中的模型样例，对 UML 模型及相关转换进行介绍。

如下是 EA 中 UML 的模型框架：

图　　示	元素说明
	静态结构图： • 类图 • 组件图 • 包图 • 复合结构图 • 对象图 • 部署图 动态行为图： • 活动图 • 通信图 • 交互概览图 • 顺序图 • 状态机图 • 时间图 • 用例图

2.1 静态结构图

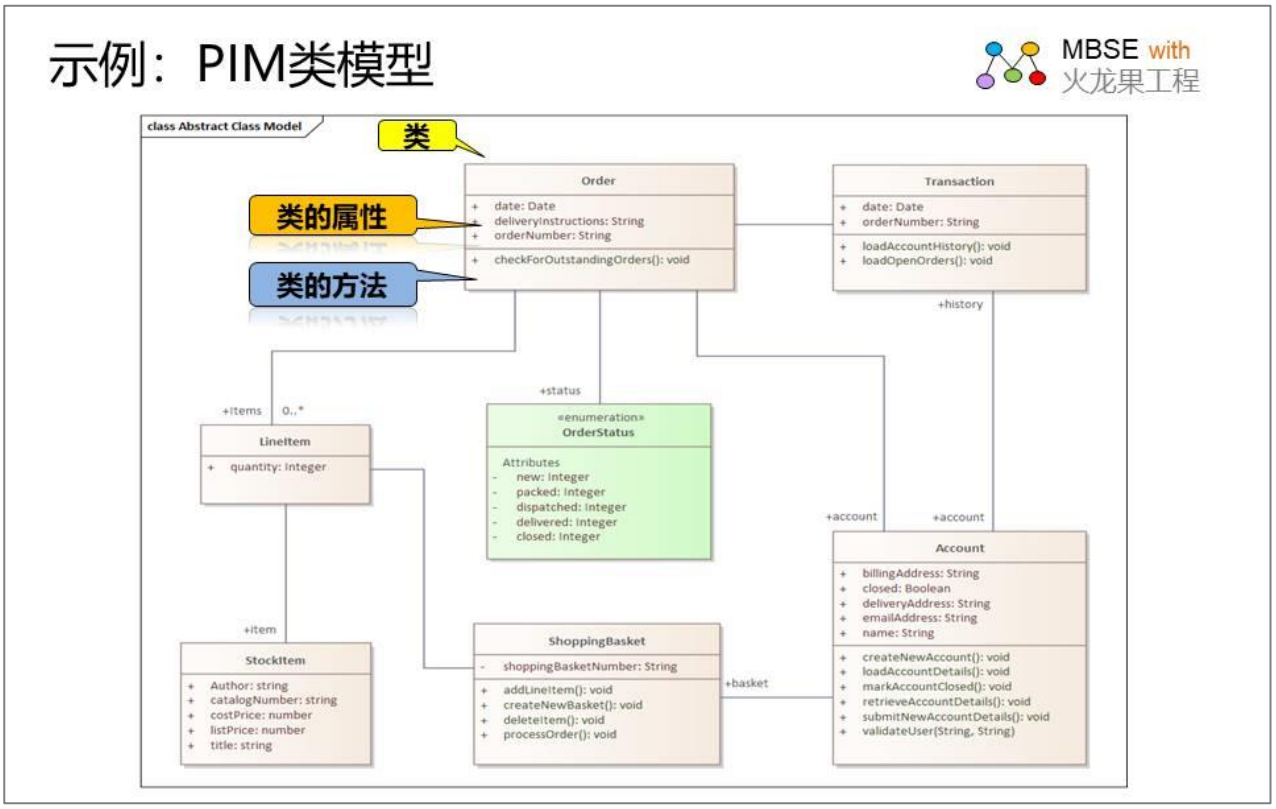
2.1.1 类图模型示例（Class Diagram）

类图是最常见的 UML 图之一，它独立于开发语言，用于从业务领域模型到编程源代码模型的结构和语义关系建模。

在看例子之前先说两个概念：PIM 类模型和 PSM 类模型。

- PIM 类模型：与平台（或者说与开发语言）无关的模型
- PSM 类模型：确定开发语言的模型

如下是一个订单系统的 PIM 类模型，它由类和类之间的关系组成：



元素图示	示例名称
类	
StockItem + Author: string + catalogNumber: string + costPrice: number + listPrice: number + title: string	Order LineItem StockItem ShoppingBasket Account Transaction
枚举	
«enumeration» OrderStatus Attributes - new: Integer - packed: Integer - dispatched: Integer - delivered: Integer - closed: Integer	OrderStatus
关系	
	（关联）

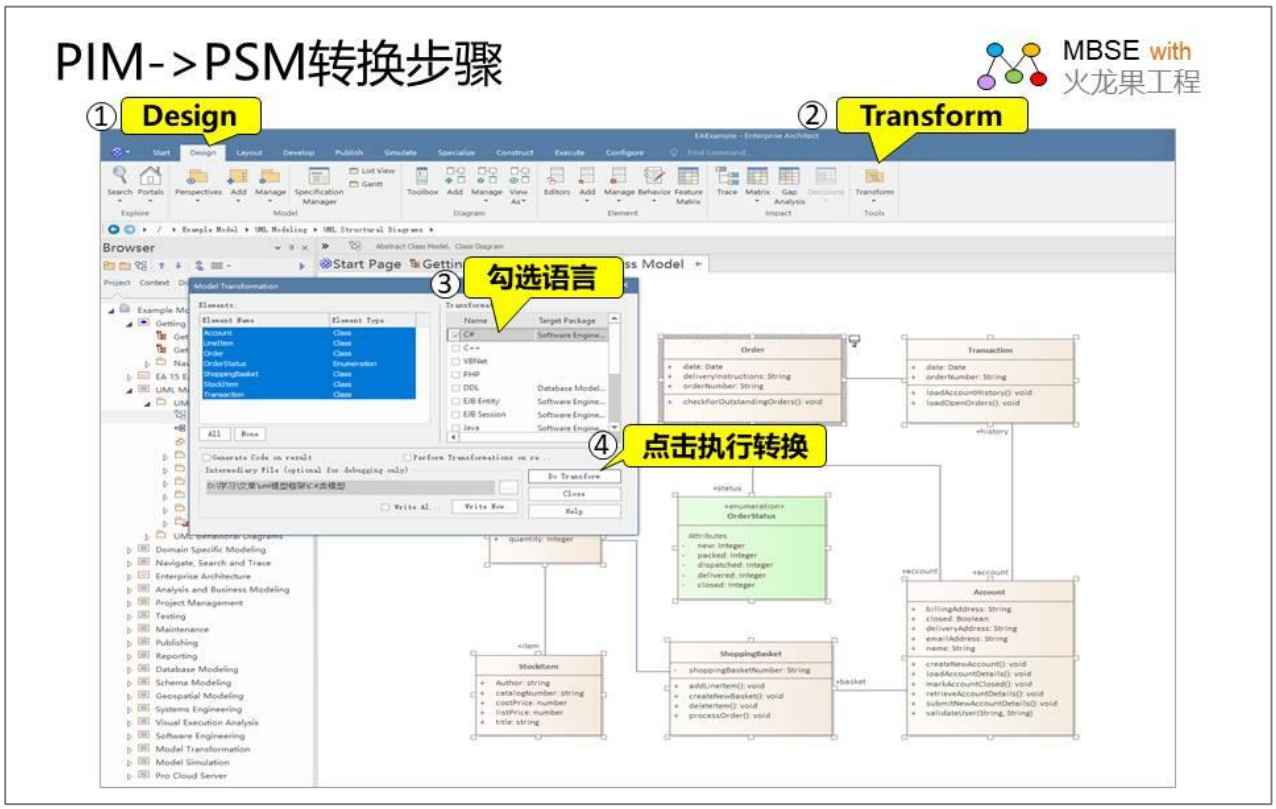
下边把这个 PIM 类模型转换成 PSM 类模型。

EA 可转换的代码类型有：

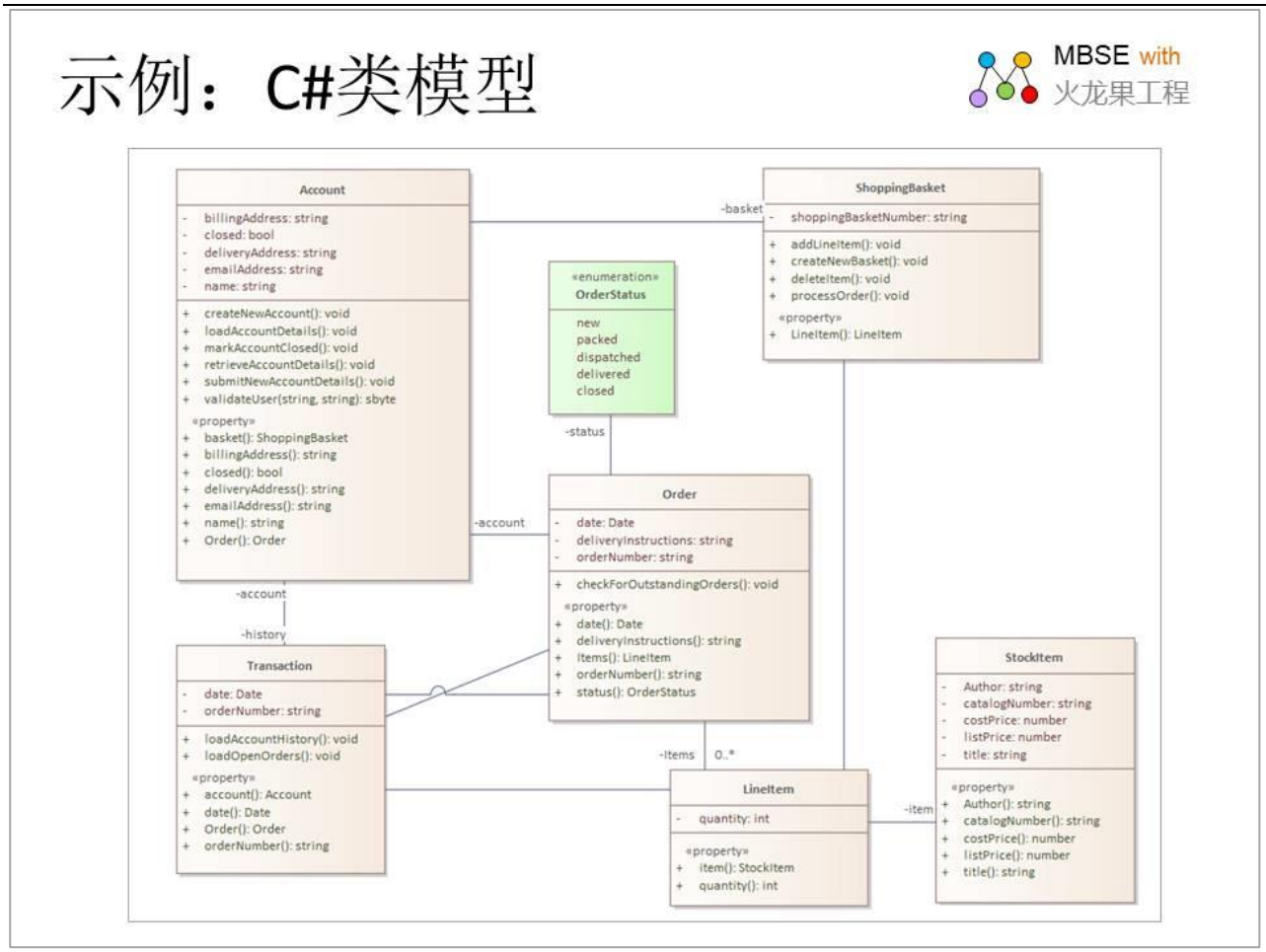
- C#
- C++
- Java

- PHP
- VB.Net

这里选择转换成 C#模型，操作步骤如下：
选中要转换的类图，然后在主菜单中选择 Design -> Transform -> Model Transformation，弹出 Model Transformation 选项框->勾选 C#，点击“Do Transform”执行转换，如图：

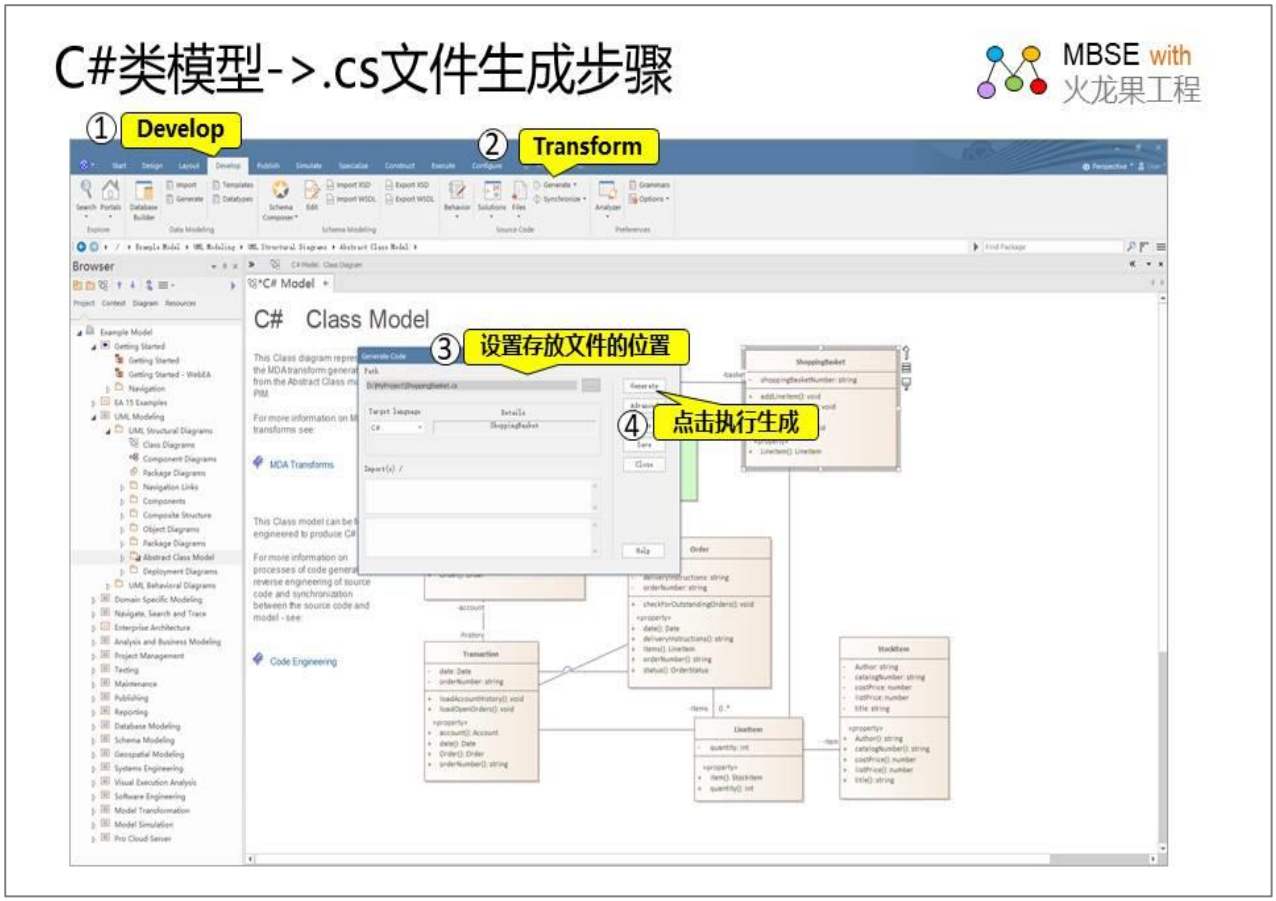


转换完成后即生成 C#类模型：

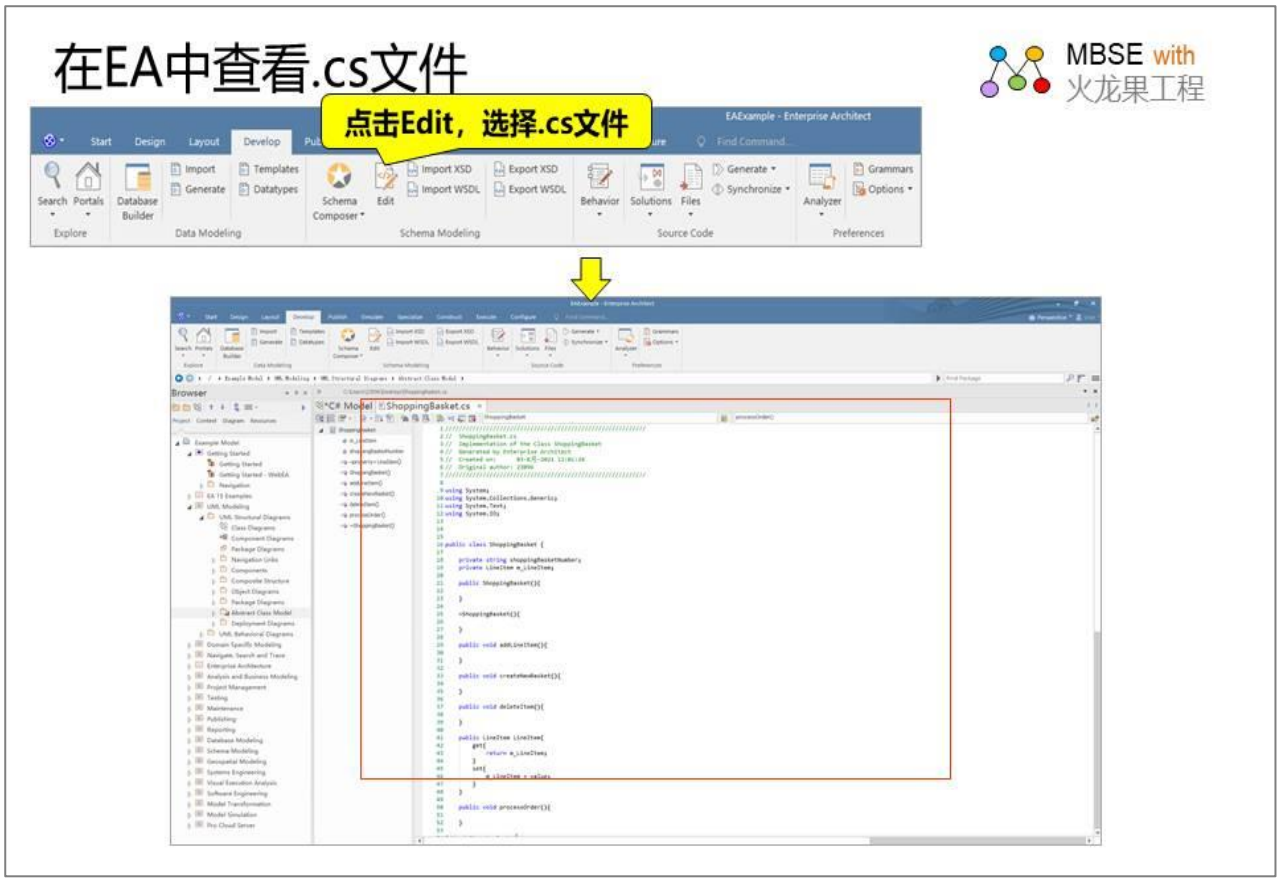


从这个模型可以得到 C#代码，操作步骤如下：

选中要转换的 C#类图，然后在主菜单中选择 Develop -> Generate ，弹出 Generate Code 操作框->设置存放 C#文件的位置，点击“Generate”，即生成.cs 文件。生成步骤及结果如图展示：



在 EA 中点击“Edit”打开刚才生成的.cs 文件，效果如下图：



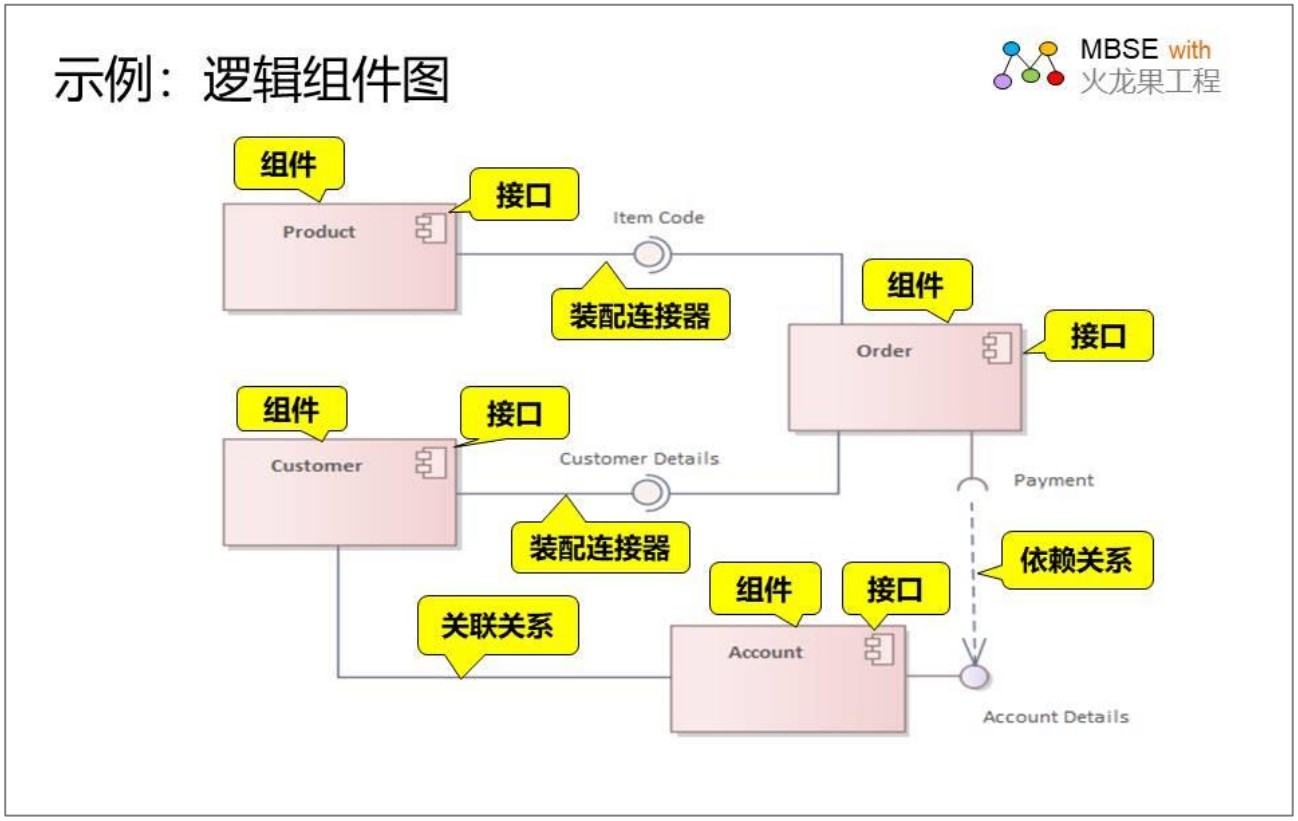
2.1.2 组件图模型示例（Component Diagram）

组件图说明了组成系统的软件、嵌入式控制器以及它们的组织结构和依赖关系，组件图比类图有更高层次的抽象。组件图通常是由一个或多个类（或对象）在运行时实现的组件集合，它们是构建块，最终一个组件图可以包含系统的很大一部分。

下边来看一下用 EA 创建的逻辑组件图和物理组件图：

逻辑组件图

逻辑组件图展示了许多组件及其相互间关系。如图：



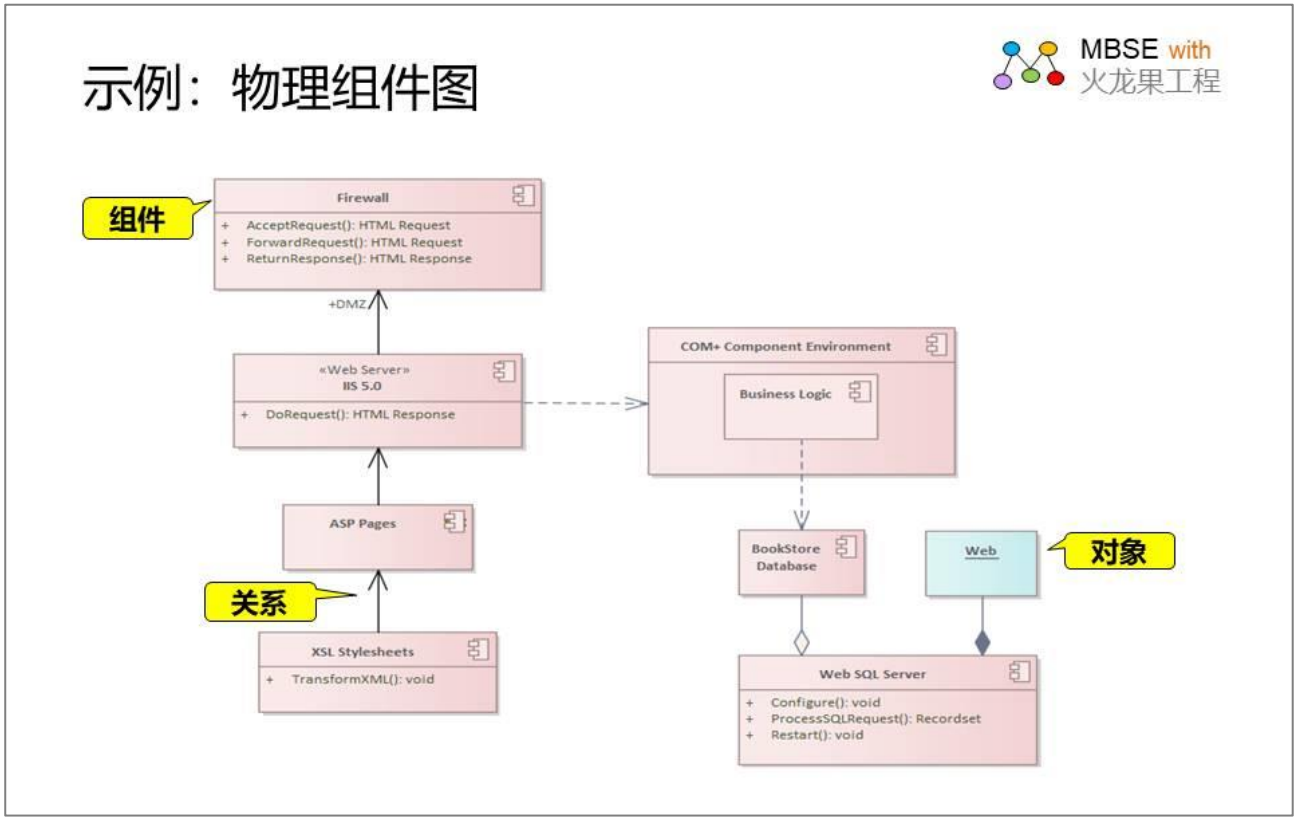
图中展示的元素有组件、装配连接器、接口和关系， 如表：

元素图示	示例名称
组件	
Product	Product Customer Order Account
装配连接器	
	Item Code Customer Detail
接口	
	Payment
关系	
	（依赖）
	（关联）

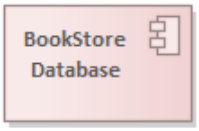
装配连接器 Item Code、Customer Detail 将组件 Product 和 Customer 提供的接口连接到 Order 指定的接口，用依赖关系将 Customer 关联的组件 Account 的详细信息映射到所需的接口 Payment（这个接口也由 Order 指定)。

物理组件图

物理组件图描述了组成系统的许多物理组件（服务器）之间的关系。如图：



图中展示的元素有组件、组件间的关系、对象， 如表：

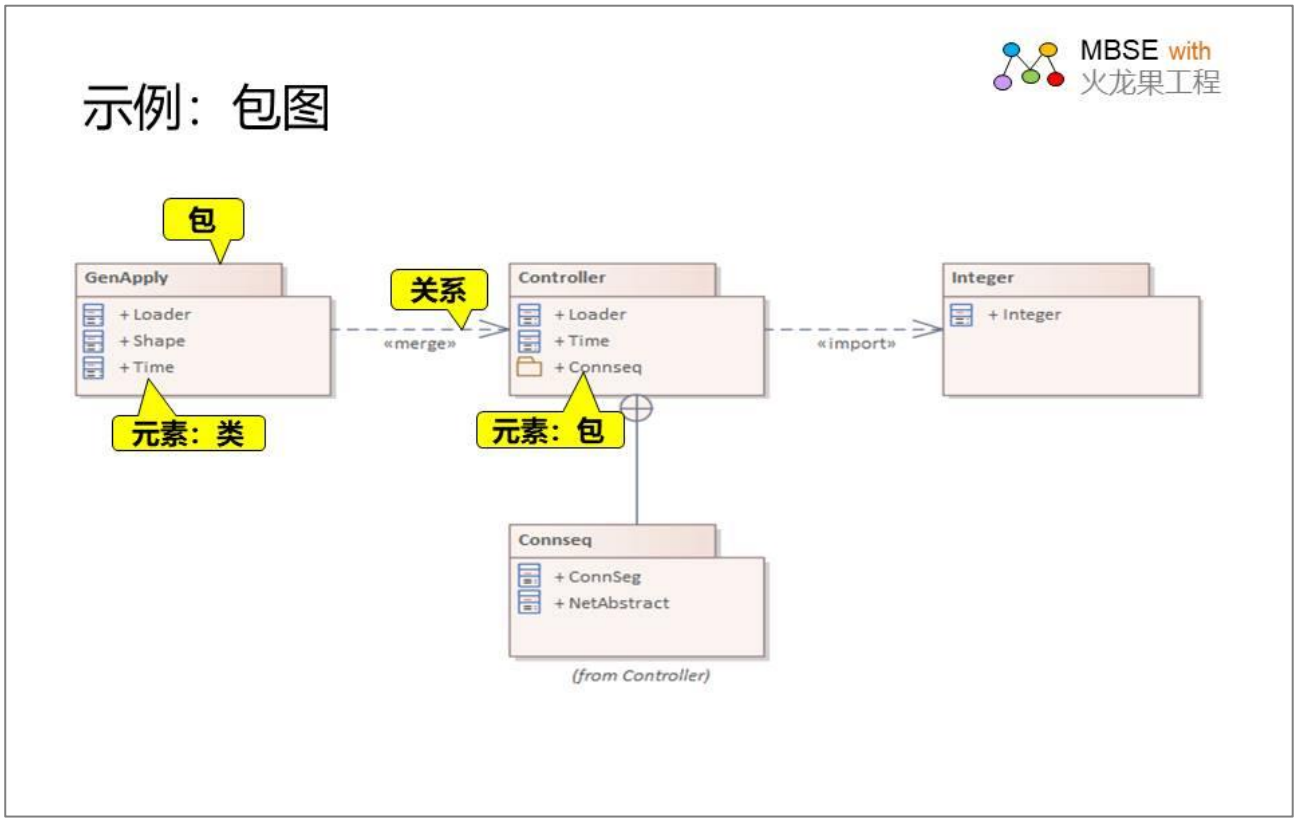
元素图示	示例名称
组件	
	Firewall Web Servers: IIS 5.0 ASP Pages XSL Stylesheets COM+ Component Environment（involves Business Logic） BookStore Database Web SQL Server
关系	
	关联
	依赖

	聚合
对象	
	Web

组件 XSL Stylesheets、ASP Pages、Web Servers、Firewall 间的关系依次为关联；组件 Web Servers 和 COM+ Component Environment 之间是依赖关系；COM+ Component Environment 与 BookStore Database 之间是依赖关系；BookStore Database、Web 跟 Web SQL Server 是聚合关系。

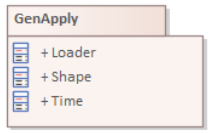
2.1.3 包图模型示例（Package Diagram）

包图描述了包中模型元素的组织以及它们之间的依赖关系，包图还提供了相应名称空间的可视化。包由多个类组成，也可以由类和包组成，如下图：



图中包含的元素有：4 个包，其中包 Controller 由类和包 Connseq 组成，其他三个包由类组成。

元素图示	示例名称
包	

	GenApply Controller Connseq Integer
关系	
	（合并）
	（导入）
	（聚合）
对象	
	Web

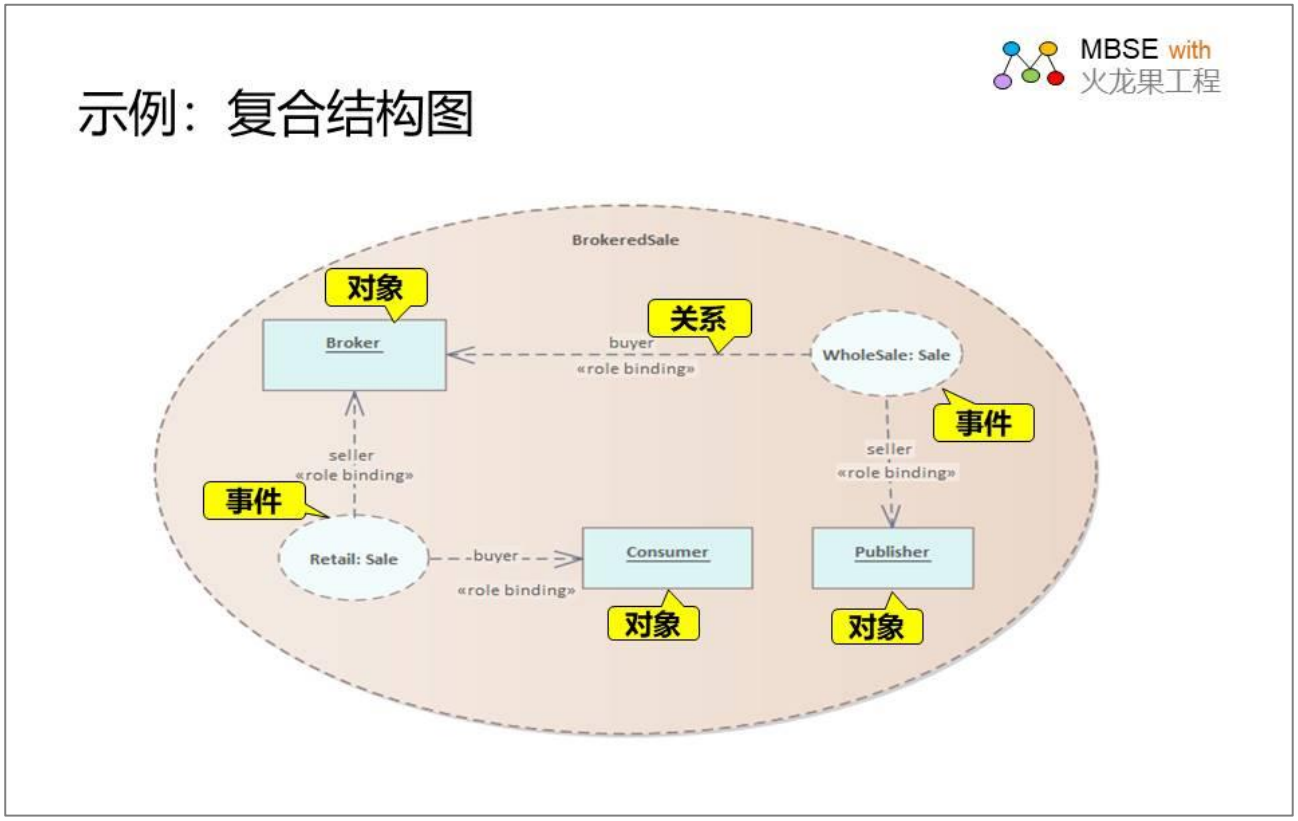
包 GenApply 与包 Controller 是合并关系；包 Controller 与包 Integer 是导入关系；包 Connseq 与包 Comtroller 是聚合关系。

2.1.4 复合结构图模型示例（Composite Structure Diagram）

复合结构图反映了类的内部协作，描述功能的接口或组件(及其属性)。复合结构图类似于类结构图，但是类结构图建模了类结构的静态视图，包括它们的属性和行为；而复合结构图为结构的特定用法建模。在复合结构图中，类作为部件或执行特定角色的运行时实例被访问。

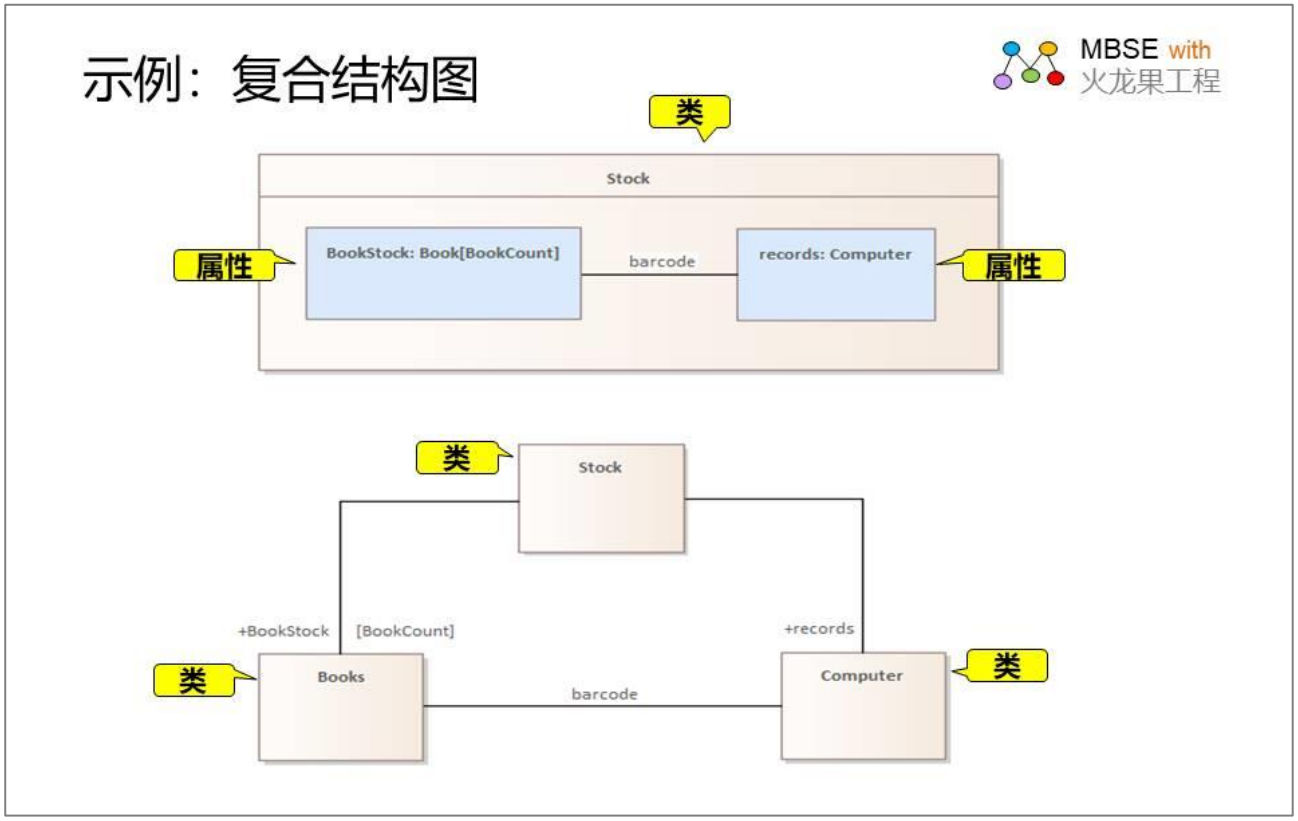
如果由类填充的角色需要多个实例，则这些部分可以具有多样性。部件类定义的端口应该在复合结构中表示，以便所有连接部件提供由端口指定的接口。

下边看两个示例：
第一个例子是事件间进行协作的复合结构图（代理销售“BrokeredSale”复合结构图），如下图：



这个示例描述的是：
批发“WholeSale”和零售“Retail”两个事件间的协作，涉及的对象有“Publisher”、“Broker”和“Consumer”，事件与对象之间是依赖的关系。

第二个例子是属性的复合结构图，如图：



这个复合结构图用两种方式表示 Stock 类的属性。第一种方式包含了类元素中的属性，而第二种方式使用连接器来反映属性。

图中元素说明：

元素图示	示例名称
协作事件	
WholeSale:Sale	WholeSale Retail
对象	
Broker	Publisher Broker Consumer
属性	
records: Computer	BookStock Records
类	
Stock	Stocks Books Computer

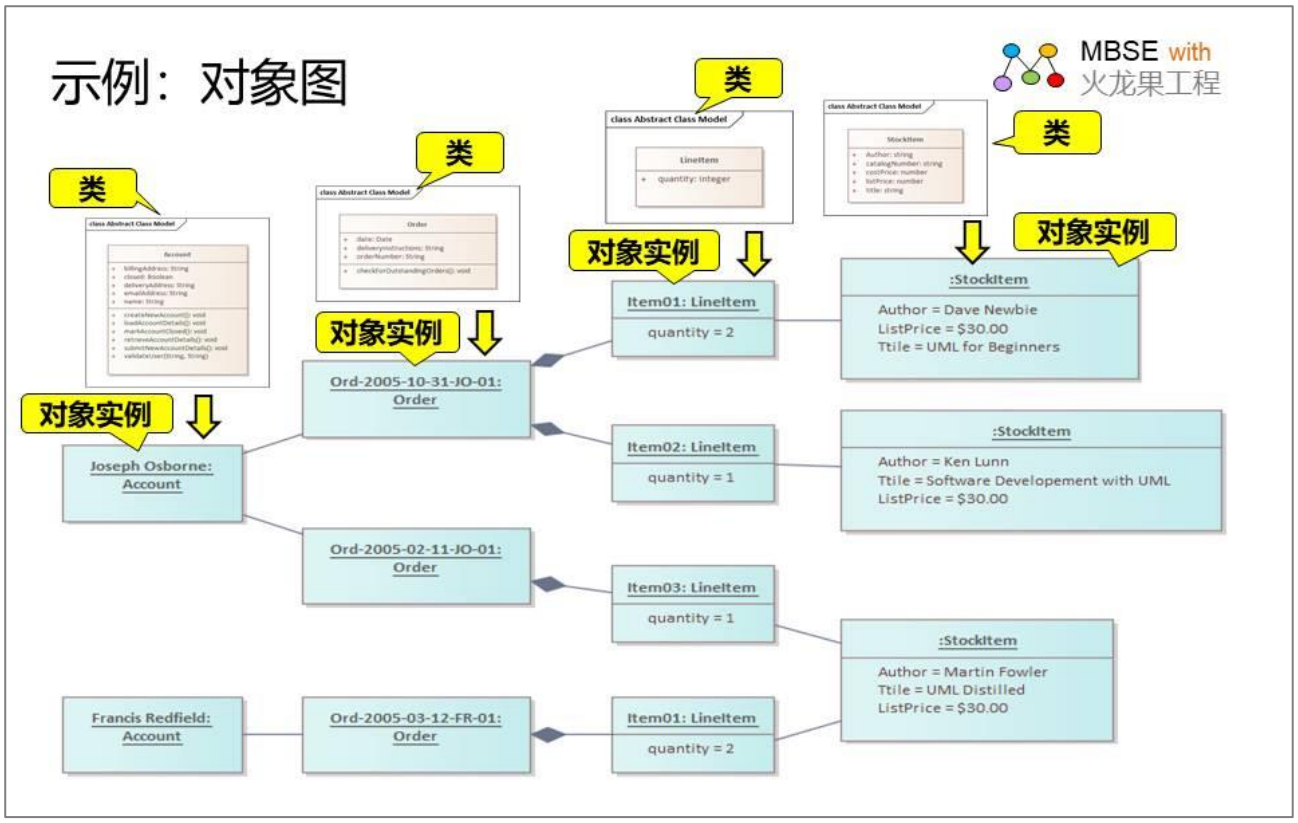
2.1.5 对象图模型示例（Object Diagram）

对象图与类图密切相关。区别在于对象图描述了类的对象实例及其在某个时间点上的关系。

对象图不显示其所对应的类图的结构变化，但反映多样性，并能为角色实例化的类服务。创建应用关系和类的不同实例，对理解复杂的类图是很有用的。

对象图也对运行时的行为建模，类似于复合结构图。不同之处在于对象图是静态类图的例证，而复合结构图反映的运行时的架构不同于静态结构图所呈现的情况。

对象图也是一种通信图(它也对对象之间的连接进行建模，并沿着每个序列事件路径)。
如下示例：



这个示例显示了来自 PIM 类模型的一组实例，它说明了两个账户持有人所做的购买活动。

类	类的对象实例
Account	Joseph Osborne Francis Redfield
Order	Ord-2005-10-31-JO-01 Ord-2005-02-11-JO-01 Ord-2005-03-12-FR-01
LineItem	Item01 Item02

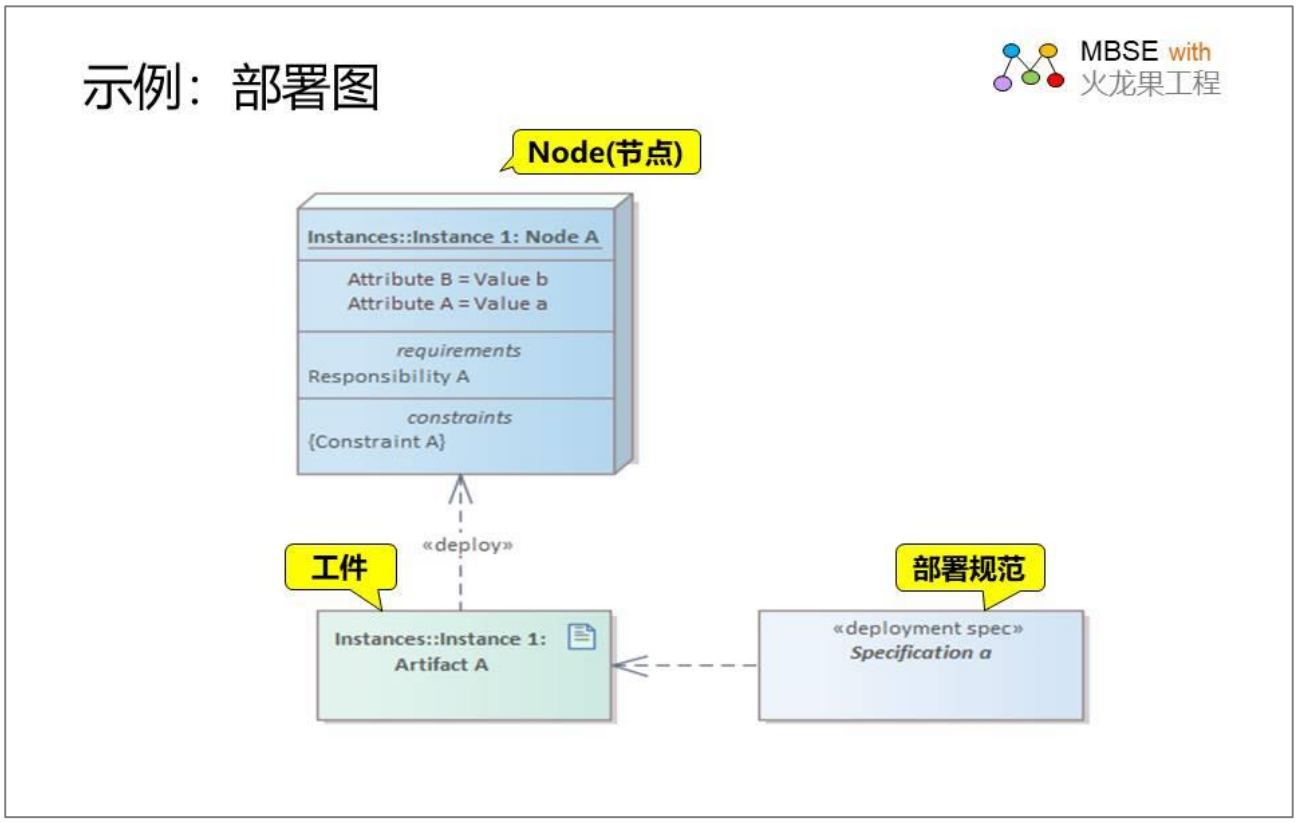
	Item03
StockItem	Item01 Item02 Item03

2.1.6 部署图模型示例（Deployment Diagram）

部署图显示系统将如何部署以及部署在哪里，即系统的执行架构。

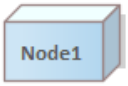
- 硬件设备、处理器和软件执行环境（系统构件）反映为节点，内部结构可以通过嵌入或嵌套节点来描述。
- 部署关系指示工件的部署，而清单关系显示组件的物理实现。
- 当工件被分配到节点进行建模系统的部署时，使用部署规范来指导分配。
- 部署图还可以指示节点具有的状态，或显示具有状态的实际运行时值的节点实例，以表示特定的条件或场景。

如下图：



图中的元素说明：

元素图示	示例名称
结点	

	Instances::Instsance1:Node A
工件	
	Instances::Instsance1:Artifact A
部署规范	
	Specification a

2.2 动态行为图

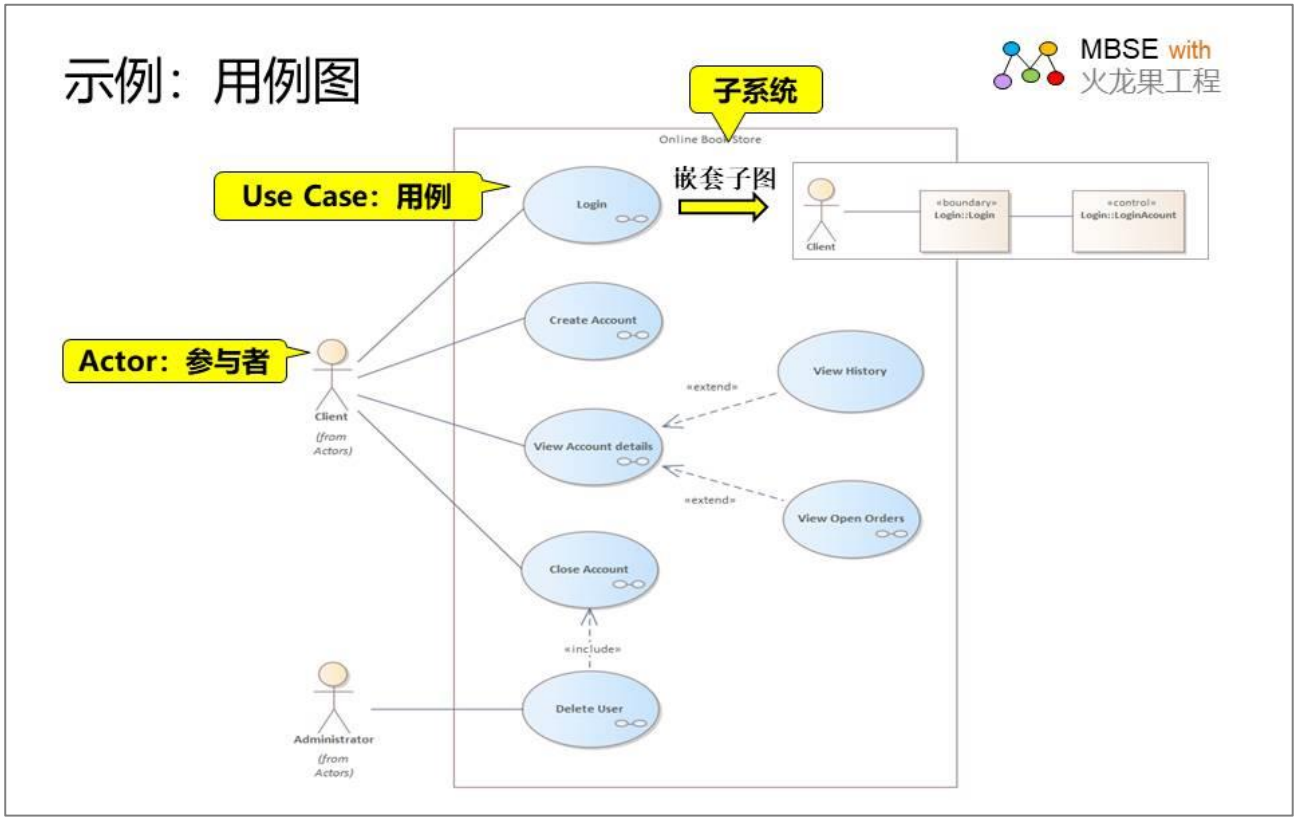
2.2.1 用例图模型示例（Use Case Diagram）

用例图捕获用例（Use Cases）以及参与者（Actors）和子系统之间的关系，可以用来：

- 描述系统的功能需求
- 描述外部事物(参与者)在系统边界上交互的方式
- 描述系统的响应

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；
参与者：是与应用程序或系统进行交互的用户、组织或者外部系统；
子系统：一般由几个用例组成，表示系统的一部分功能。

下面是在线书店用户管理子模块的类图示例：



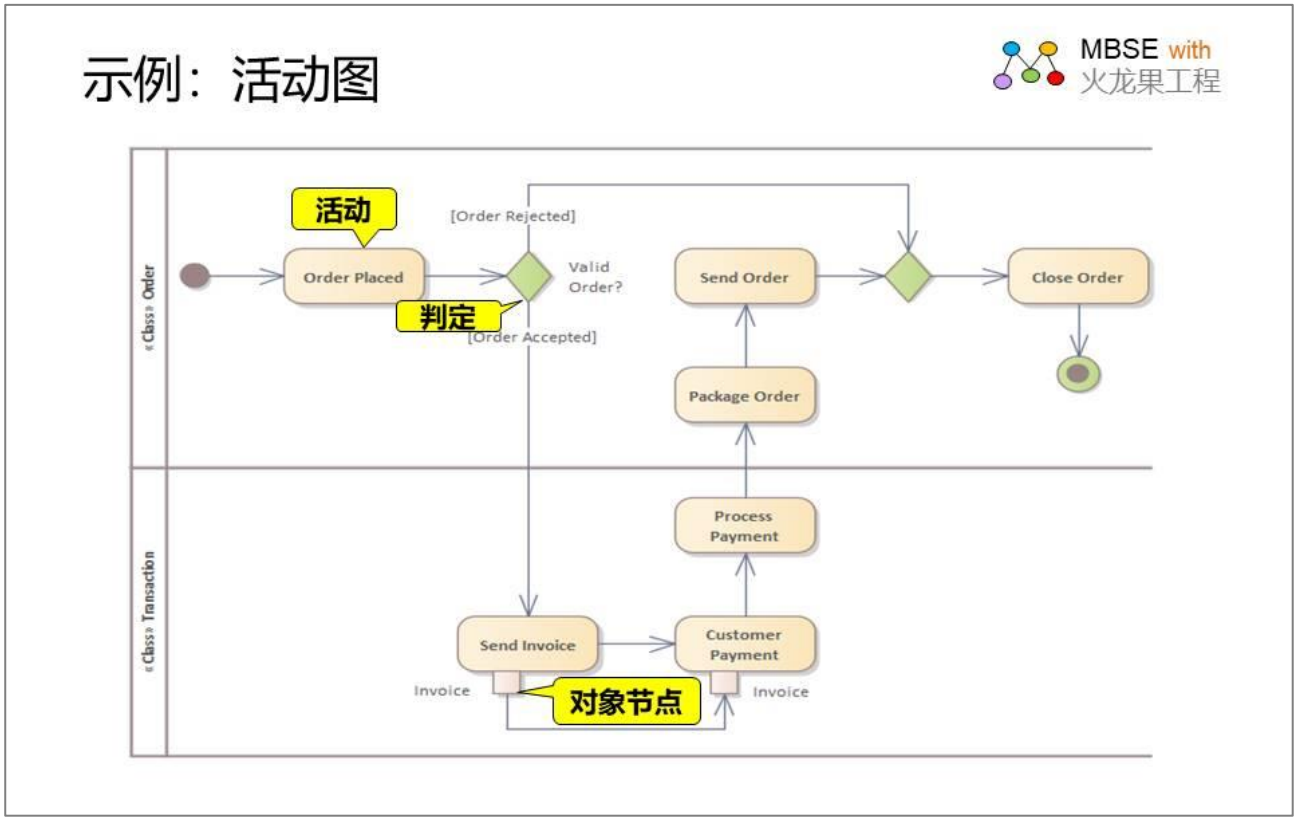
图中的元素说明：

元素名称	示例名称
子系统	Online Book Store
参与者（角色）	Client Administrator
用例	Login Create Account View Account details Close Account View History View Open Orders Delete User

- 子系统：Online Book Store-用户管理子系统
- 用户 《Client》 的 UseCase 有”Login”、 “Create Account”、”View Account details”、 “Close Account”；”View Account details”扩展出了”View History”和”View Open Orders”。
- 用户《Administrator》的 UseCase 有” Delete User”，”Delete User”包含”Close Account”.
- 用例包含嵌套的序列图和通信图，这些子图描述用例的基本流。如图上示例的是顾客登录的基本流。

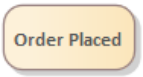
2.2.2 活动图模型示例（Activity Diagram）

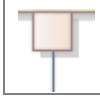
活动图用于对系统行为建模，说明系统的各个元素之间的交互。它们显示了系统遵循的基于条件测试、并发处理、数据访问和过程中断的逻辑路径。可以描述业务流程、程序流程、用户操作流程。如下图所示：



此图显示了使用泳道来组织元素，这些元素可以水平或垂直方向放置。该图还显示了通过对象流连接的输出和输入引脚的使用

图中主要元素说明：

元素名称	示例名称
泳道	
	Transaction Order
动作	
	Order Placed Send Order Close Order Package Order Process Payment Customer Payment Send Invoice

判定	
	Valid Order?
动作指针	
	Invoice

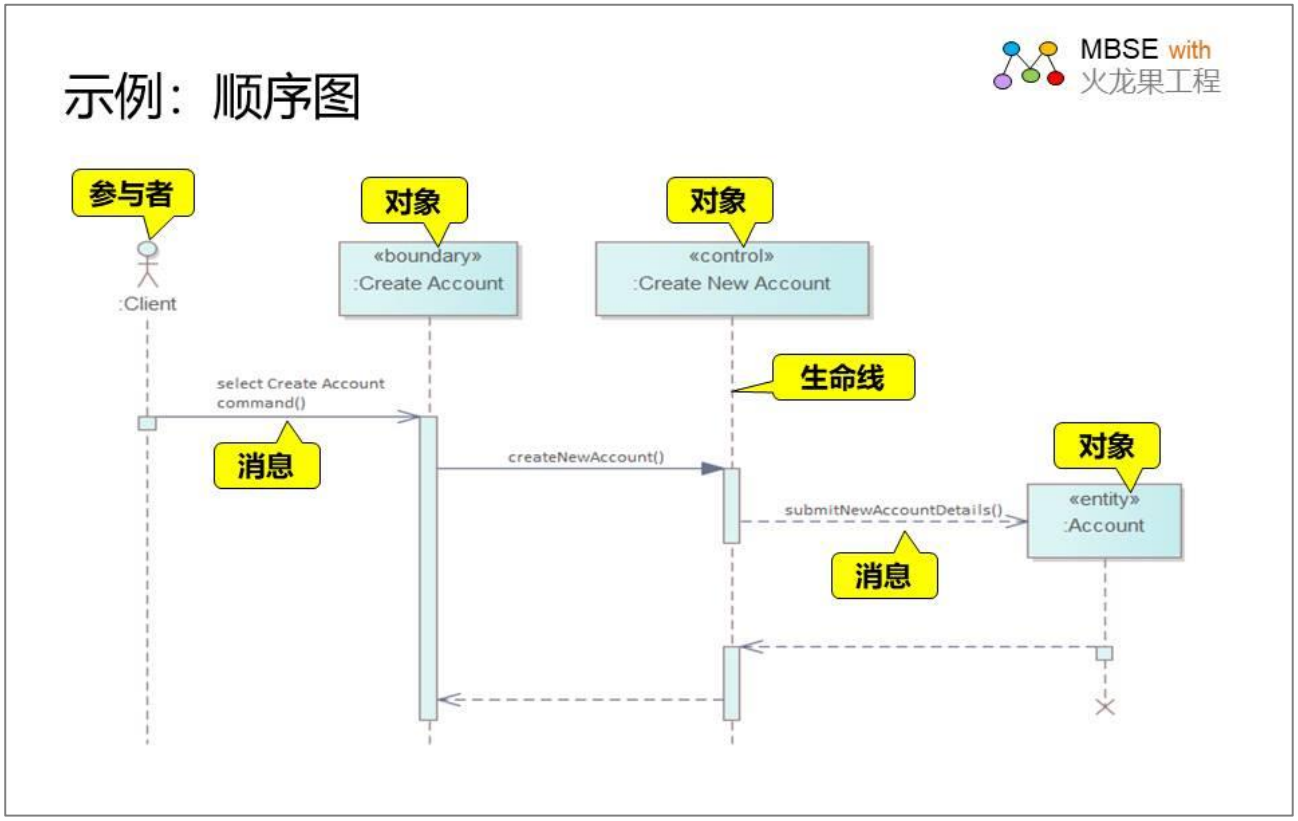
2.2.3 顺序图模型示例（Sequence Diagram）

顺序图描述几个对象随时间顺序进行的交互。通过创建包含参与者和用例中涉及的元素的顺序图，为用户和系统要完成的任务的步骤进行顺序建模。

顺序图可以：

- 描述工作流、消息传递以及元素如何随着时间的推移而协作；
- 在分析的早期阶段，获取整个系统的信息流和职责；
- 元素之间的消息最终成为 Class 模型中的方法调用；
- 为用例场景制作解释性模型；

见如下示例，描述了用户注册账户涉及的对象及对象间的消息交互的顺序，图中参与者及对象元素带有生命线：

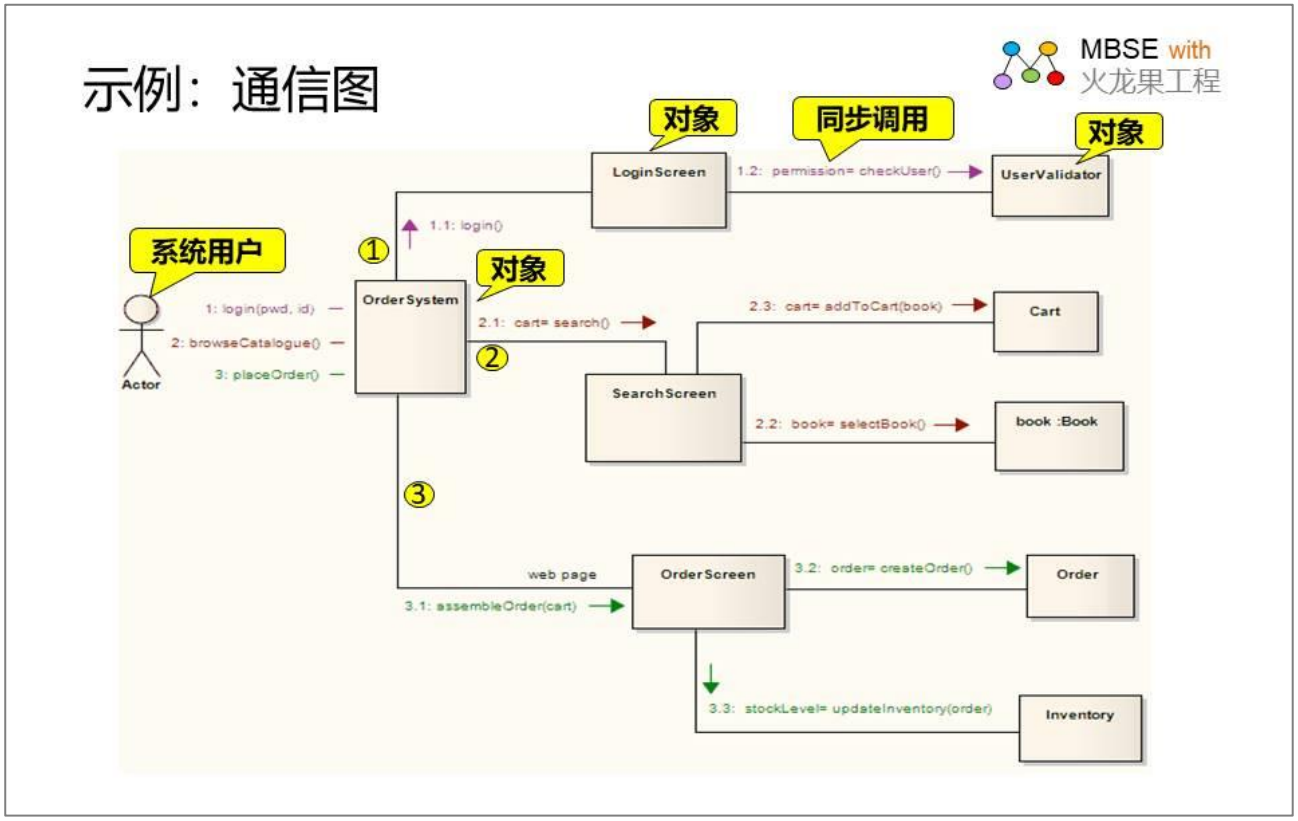


2.2.4 通信图模型示例（Communication Diagram）

通信图描述几个对象的交互通信关系，更关注对象之间的连接关系而非交互顺序；它可以描述用户和系统的交互，也可以描述程序对象之间的交互。交互消息可以覆盖在类或组件图上，以显示元素是如何进行通信的。

通信图使用有序的标记关联来说明处理过程。采用编号指示处理的顺序和嵌套。编号样式可以是：1　1.1　　1.1.1　　1.2 等等。

见如下图书订购的通信图示例：
用户操作订单系统，首先进行用户登录验证，然后进行图书查询，最后生成订购单。

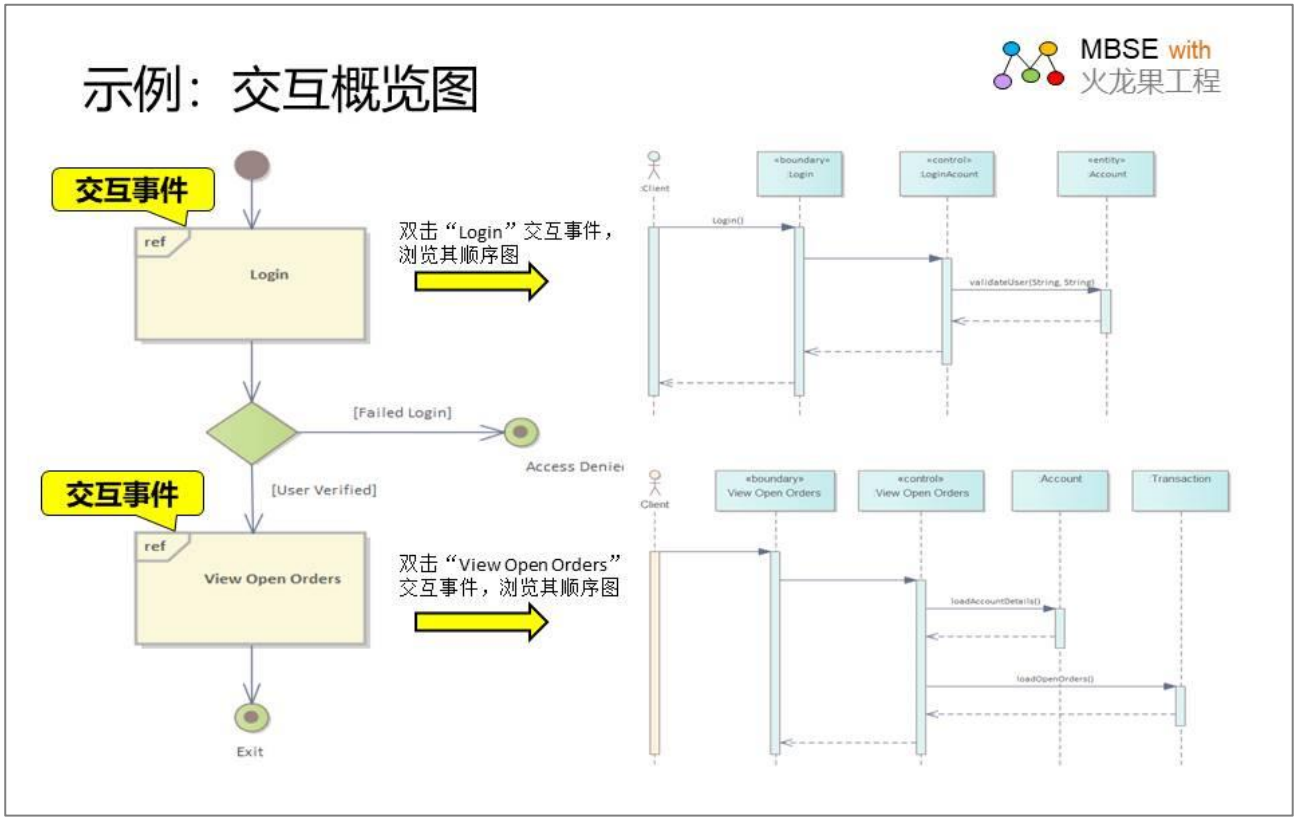


1：login(pwd,id)
  1.1：Login()
  1.2：permission=checkUer()
2：browseCatalogue()
  2.1：cart=search()
  2.2：book=selectBook()
  2.3：cart=addToCart(book)
3：placeOrder()
  3.1：assembleOrder(cart)
  3.2：order=createOrder()
  3.3：stockLevel=updateinventory(order)

2.2.5 交互概览图模型示例（Interaction Overview Diagram）

交互概览图用于描述在整个控制流的过程中，各个交互场景中对象的交互，类似于高级活动图。因此，它使用称之为“片段”的交互事件（InteractionOccurrence）来表示一系列活动，可以通过双击“片段”查看这些活动的细节。

见如下浏览订单的交互概览图示例：



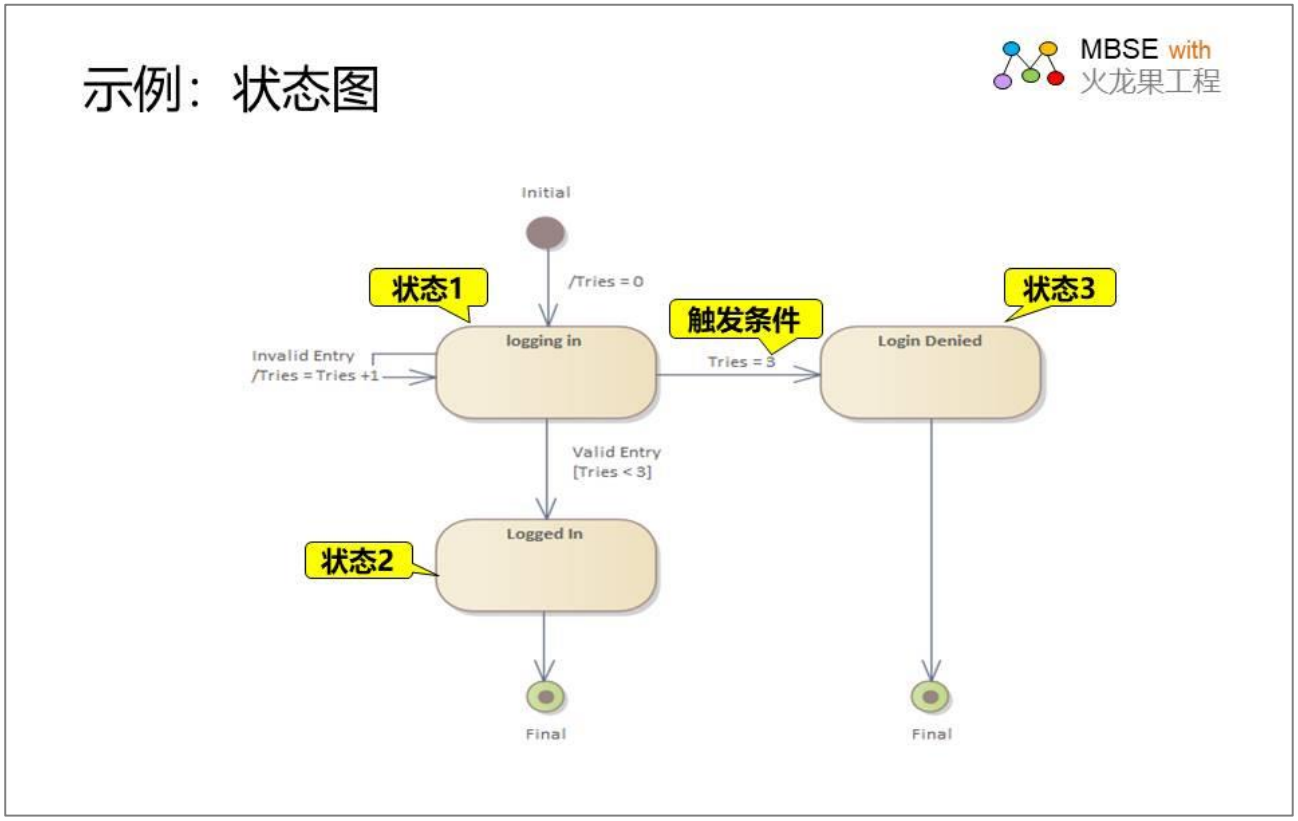
这是用户登陆系统查看订单的交互概览图，包含两个交互事件：

- Login 事件
  双击“Login”打开用户登录的顺序图（如右上图），查看登录过程中涉及的三个对象“《boundary》:Login、《control》:LoginAccount、《entity》:Account”如何进行交互；
- View Open Order 事件
  双击“View Open Order”打开用户查看开放订单的顺序图，了解登录过程中涉及的四个对象“《boundary》:View Open Orders、《control》: View Open Order、:Account”、:Transaction”如何进行交互。

2.2.6 状态图模型示例（StateMachine Diagram）

状态图用于描述一个对象(通常是 Class 或者 Actor)如何在一个生命周期中的各状态之间移动，并根据转换触发器和约束警示条件对其行为进行分类。

见如下“登录”的状态图示例：
“登录”一共有 3 个状态：进行登录（logging in）、登录成功（Logged in）、拒绝登录（Login Denied）。

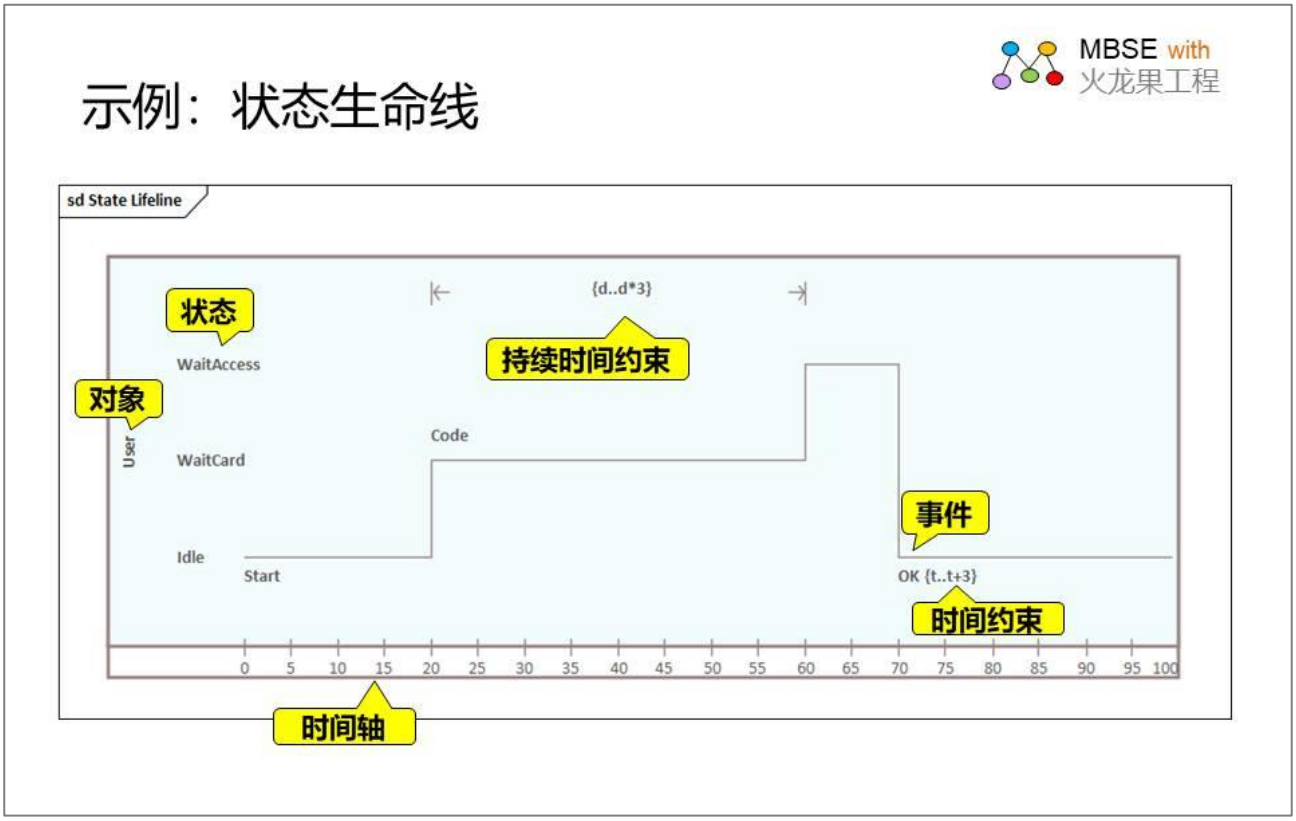


2.2.7 时间图模型示例（Time Diagram）

时间图描述了在时间尺度内不同对象随时间变化的状态和交互。它可以在定义或分析通信协议、服务器响应、软件组件、业务或系统流程或其他实体时使用，这些实体的实际时间值对分析至关重要。与状态图相比较，状态图显示状态之间的转换序列，但不显示这些转换相对应的时间尺度。

下边结合时间图模型示例，分析时间图的两种形式：状态生命线和数值生命线。

- 状态生命线（State Lifeline）
  状态生命线表示对象在时间度量范围内的状态。X 轴表示时间，y 轴表示状态之间的离散转换。



状态转换可以用属性进行描述，常用的属性有：时间约束、持续时间约束、事件等，说明如下：

属　性	描　述
Time Constraint 时间约束	指在生命线内更改状态所花费的时间，或传输消息所花费的时间。
Duration Constraint 持续时间约束	描述在一个特定状态下的生命线周期，约束条件由生命线内的状态更改或生命线收到消息引起。
Event 事件	描述正在发生的事件。
At Time 时间	指定状态改变的开始时间。
Transition to 转变至	指示生命线更改到的状态。

示例图中 OK 转换点具有以下属性：

属 性	值
At Time	70ms
Transition to	Idle
Event	OK
Time Constraint	t..t+3

- 数值生命线（Value Lifeline）

数值生命线描述生命线的状态，平行线表示稳定的状态，线之间的交叉表示状态的转变。

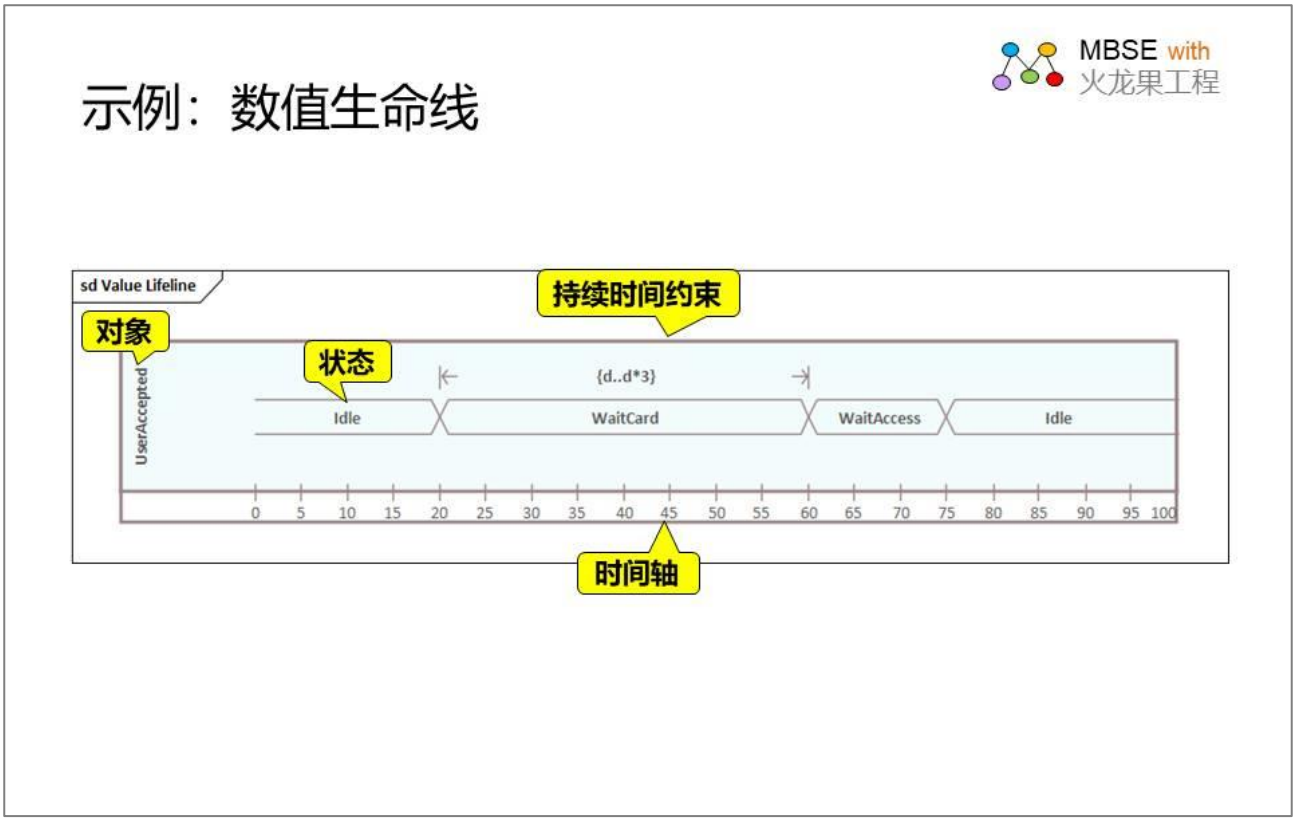
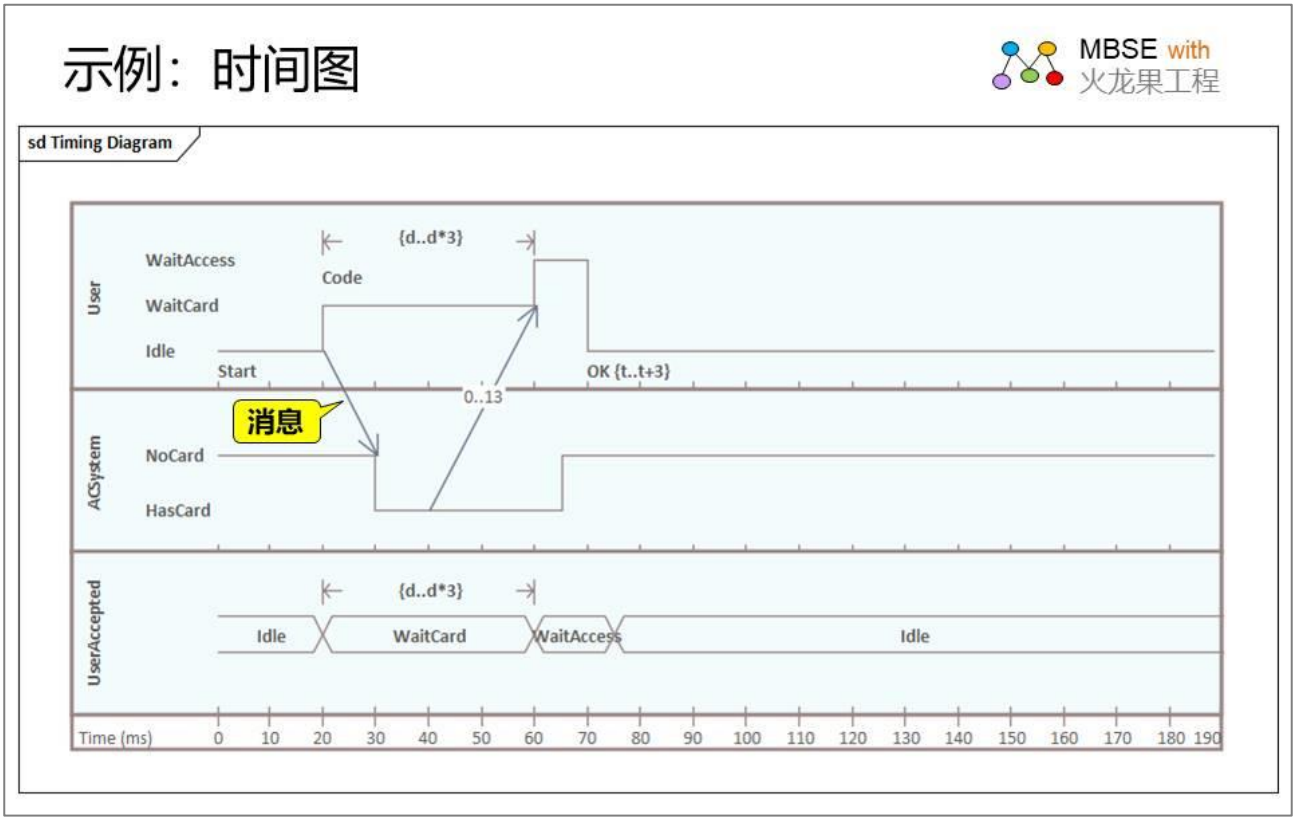


图 10ms 的转换点具有以下属性：

属 性	值
At Time	20ms
Transition to	Waitcard
Event	Switch
Duration Constraint	d..d+3

当状态生命线和数值生命线的 X 轴即时间轴相同时，可以把两者叠加。上边两图叠加后的效果如下：



图中的箭头表示消息（Message），是时间图的属性之一：

属　性	描　述
Message 消息	指在生命线内更改状态所花费的时间，或传输消息所花费的时间。

以上介绍了 EA 中 UML 的模型框架，在实际应用的时候，需要一定的时间熟悉各种模型的元素、关系等等。欢迎交流！

本文的图例采用了建模工具EA　ENTERPRISE ARCHITECT

如果您希望了解更多信息：

- 欢迎访问建模者频道 http://www.sysml.org.cn/
- 也欢迎直接联系我们 zhgx@uml.net.cn ，010-62670969

下载 pdf 版：《UML 模型框架（基于 EA）》

本文使用的建模工具为 EA，可以下载试用版 http://tool.uml.com.cn/ToolsEA/download.asp

MBSE with 火龙果

希望您读了此文后有所受益。
如果您有经验乐于分享，欢迎投稿给我们。
如果您对我们的培训、咨询和工具感兴趣：

课程：基于 UML 和 EA 进行分析设计
课程：MBSE（基于模型的系统工程）
课程：基于模型的需求管理)方法与实践
课程：基于 SysML 和 EA 进行系统设计与建模
课程：企业架构建模
课程：系统架构建模方法与案例
课程：领域驱动的建模与设计
课程：基于模型的设计
课程：业务建模与业务分析
建模工具：EA
基于模型的需求管理工具：iSpace
咨询方案：MBSE(基于模型的系统工程)
咨询方案：基于 UML 的模型驱动的开发