

## 实验 4 面向对象的系统分析与设计（二）

### 1 实验目的

熟悉使用 Enterprise Architect 工具绘制分析模型，并学会对实际案例进行用例分析。具体包括：

- (1) 定义系统备选架构；
- (2) 识别分析类；
- (3) 构造用例实现-绘制顺序图；
- (4) 构造用例实现-绘制 VOPC 类图；
- (5) 完成实体类类图；
- (6) 对案例进行用例分析。

### 2 实验准备与内容

#### 2.1 实验准备

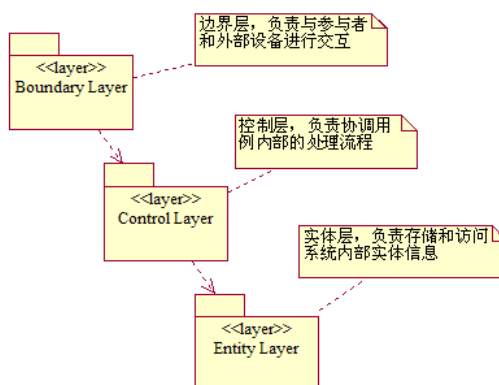
- (1) 安装好 Enterprise Architect 12 的电脑；
- (2) 实验 3 已完成的 EA 项目文件“旅游.eap”和“医院.eap”；
- (3) 课件《06.用例分析.pptx》；
- (4) 补充材料《旅游业务申请系统系统分析.pdf》；
- (5) 补充材料《医院预约挂号系统案例描述.pdf》

#### 2.2 实验内容

- (1) 练习 1 定义系统备选架构：绘制《旅游业务申请系统系统分析.pdf》图 1 中的系统备选架构图。
- (2) 练习 2 识别分析类：绘制《旅游业务申请系统系统分析.pdf》中图 2~图 4 的分析类。
- (3) 练习 3 构造用例实现-绘制顺序图：为用例“办理申请手续”的用例实现绘制《旅游业务申请系统系统分析.pdf》中图 5 的顺序图。
- (4) 练习 4 构造用例实现-绘制 VOPC 类图：为用例“办理申请手续”的用例实现绘制《06.用例分析.pptx》第 64 页的 VOPC 类图。
- (5) 练习 5 完成实体类类图：绘制《旅游业务申请系统系统分析.pdf》中图 6 的实体类类图。
- (6) 练习 6 对案例进行用例分析：对《医院预约挂号系统案例描述.pdf》中的医院预约挂号系统进行用例分析。

### 3 实验指导

### 3.1 练习 1 定义系统备选架构

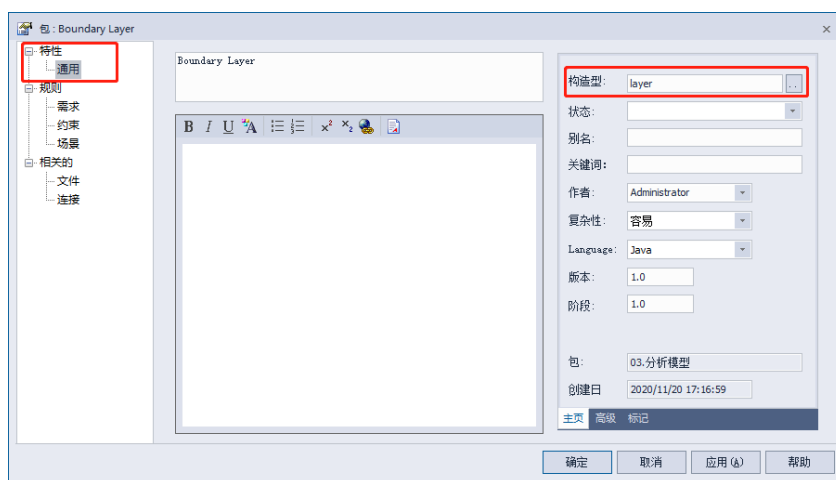


本练习的目标是在 EA 工具中对《旅游业务申请系统系统分析.pdf》图 1（上图）中的系统备选架构图进行绘制。

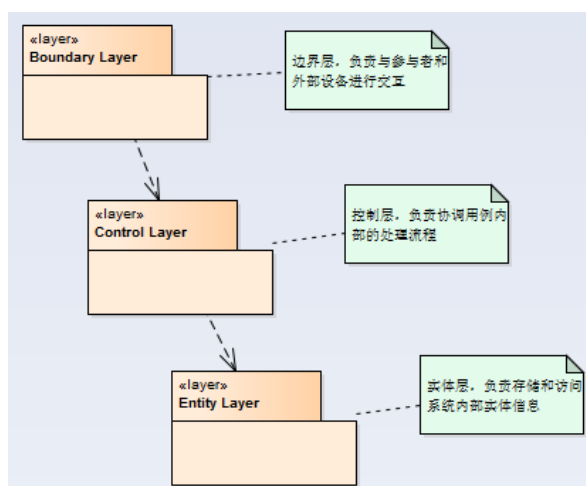
首先在 EA 工具打开项目“旅游.eap”（实验 3 已完成的项目文件）。

在“分析模型”中，新增“主视图”，“主视图”的类型选择“包图”（UML Structural – Package）。

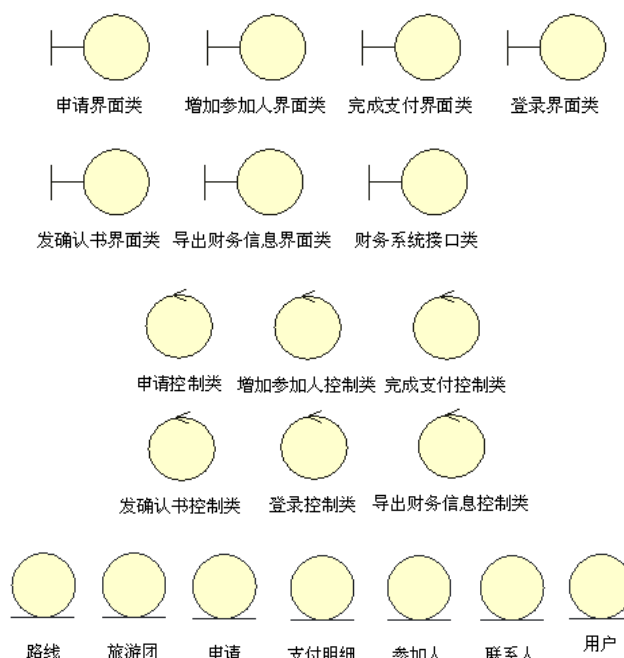
从工具箱中拖入元素“Package”，命名为“Boundary Layer”，点击“确定”后，在弹出的特性窗口中，在“特性-通用”标签下的“构造型”输入框中，输入“layer”表示其构造型为“layer”：



绘制其它部分，添加依赖关系、注释等，效果图如下：



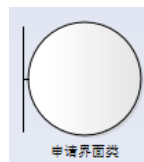
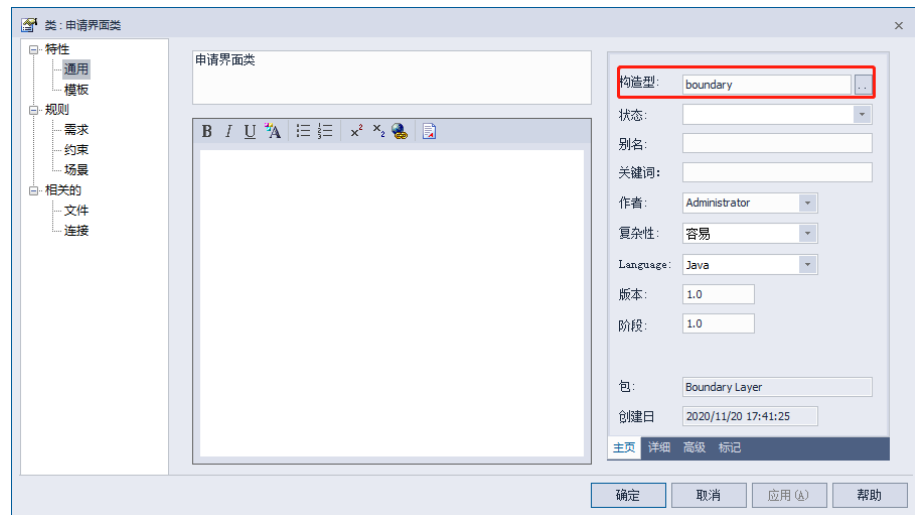
### 3.2 练习 2 识别分析类



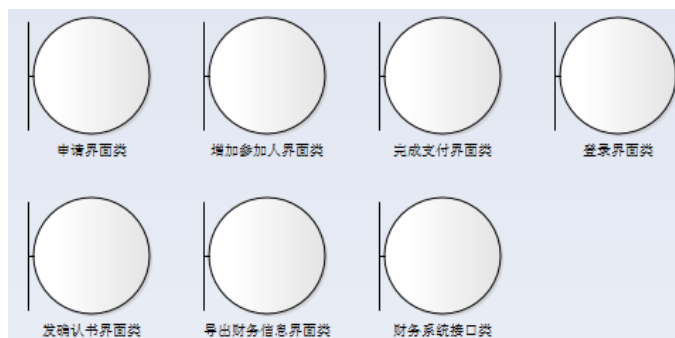
本练习的目的是在分析模型中，添加《旅游业务申请系统系统分析.pdf》中图 2~图 4（上图）的分析类。

在“项目浏览器”中，在包“<<layer>>Boundary Layer”中新增“主视图”，类型选择“类图”（UML Structural - Class）或“包图”。

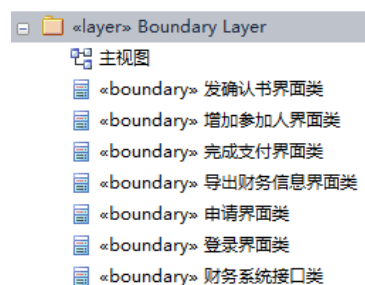
双击该“主视图”进行绘制。从工具箱中选择元素“Class”拖入绘制区域，命名为“申请界面类”，构造型输入“boundary”，点击“确定”，即可获得“申请界面类”：



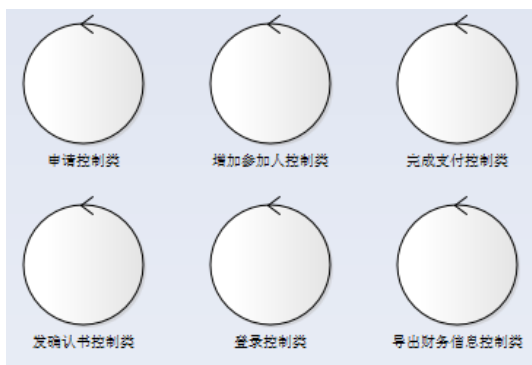
同理，增加其它边界类，效果图如下：



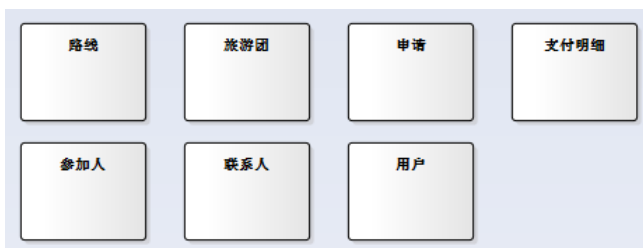
此时的“<<layer>>Boundary Layer”包的结构可参考：



按照类似方法，在控制层“<<layer>>Control Layer”中新建“主视图”，并使用构造型“control”，完成所有控制类的绘制：

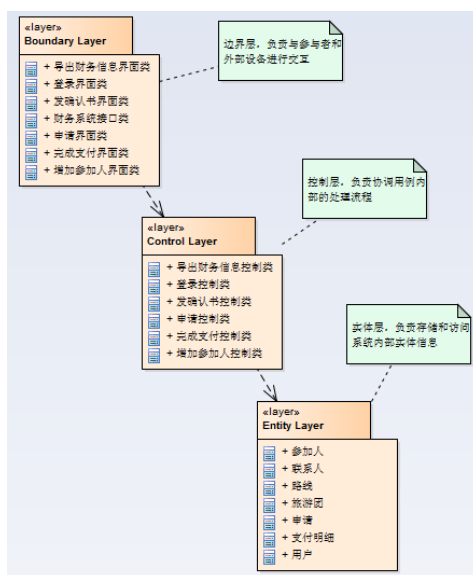


在实体层 “<<layer>>Entity Layer” 中新建 “主视图”，完成实体类的绘制：

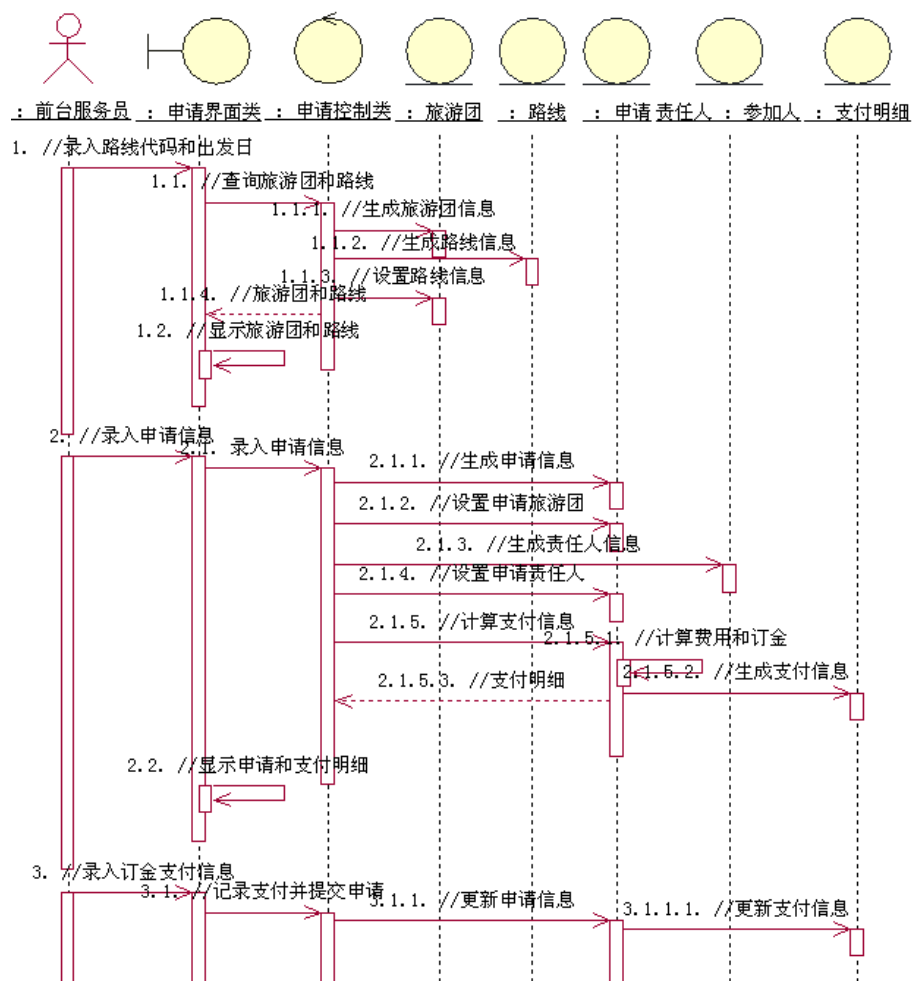


注意：在以上模型效果图中，对实体类的定义没有使用构造型 “entity”，主要考虑的原因是，当使用构造型 “entity” 时，类的属性和操作将无法在图中显示出来（这是 EA 工具的局限性）。为了使分析类类图可以表示更多的信息，因此我们在这里推荐不要使用构造型 “entity” 绘制实体类。当然，使用构造型 “entity” 也是可以的，在这种情况下也可以正常定义类的属性和操作，只是它们无法在视图中显示出来。

在完成分析类的绘制后，我们发现“分析模型”的主视图已经得到细化，如下图：



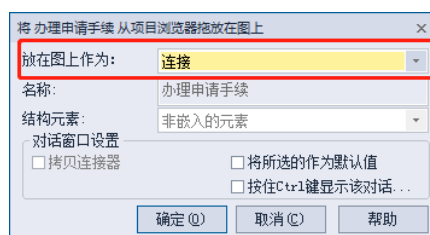
### 3.3 练习 3 构造用例实现-绘制顺序图



本练习的目的是为用例“办理申请手续”构造用例实现，并绘制《旅游业务申请系统系统分析.pdf》中图 5（上图）的顺序图。

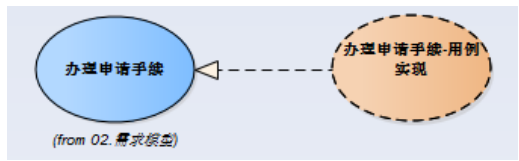
首先在“03.分析模型”中新增包“用例分析”，我们将在该包中组织用例实现。在“用例分析”包中新建图，命名为“用例分析”，类型选择“用例图”（UML Behavioral – Use Case）。双击打开该图进行用例与用例实现（协作）之间关系的绘制。

本练习只考虑用例“办理申请手续”。从“项目浏览器”中，将“需求模型”（实验 3 已绘制）里的用例“办理申请手续”拖入“用例分析”图中，在弹出的窗口中，“放在图上作为”一栏选择“连接”，表示直接引用：

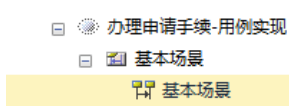


点击“确定”。

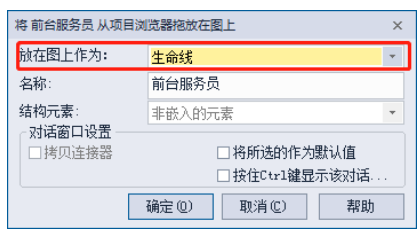
从“工具箱”中选择元素“Collaboration”（协作），放入“用例分析”图中，命名为“办理申请手续-用例实现”。使用“实现”（Realize）关系连接用例“办理申请手续”和它的用例实现：



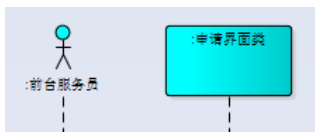
在“办理申请手续-用例实现”中增加该用例基本场景的顺序图：右键点击图中“办理申请手续-用例实现”选择“新建子图”（或在“项目浏览器”中右键点击“办理申请手续-用例实现”选择“增加”），选择“交互元素-与顺序图”，新建复合元素的名称为“基本场景”，点击“确定”后即成功新建“办理申请手续-用例实现”的顺序图：



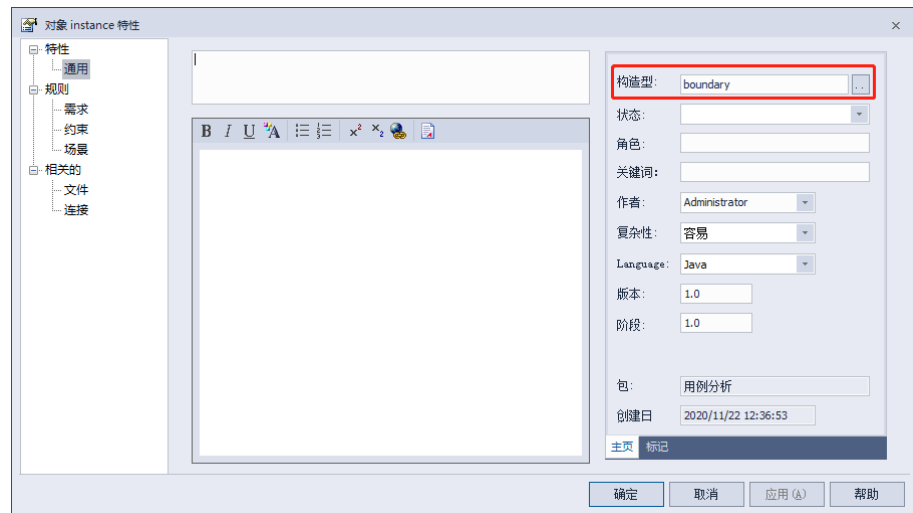
对顺序图“基本场景”进行绘制：在“项目浏览器”中，从“02.需求模型”将参与者“前台服务员”拖入顺序图“基本场景”的绘制区域内，在弹出的窗口中，“放在图上作为”一项选择“生命线”，表示这是定义了参与者“前台服务员”的一个对象/实例：



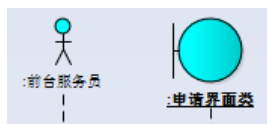
同理,从边界层“<<layer>>Boundary Layer”中拖入“申请界面类”，并选择“生命线”，得到以下“基本场景”图：



注意这些元素都是“对象”元素（绿色）。此时双击“申请界面类”对象打开特性窗口，在“构造型”中输入“boundary”：

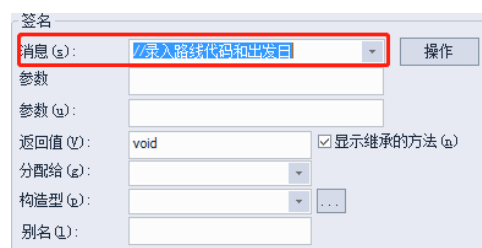


点击“确定”后，得到以构造型特殊图形表示的“基本场景”图：



类似地，利用构造型“control”和“entity”完成顺序图里其它对象的绘制。其中，图中“参加人”类的对象具有对象名称“责任人”，注意在该对象的特性窗口中进行设置。

顺序图中消息的绘制：从“工具箱”中点选“Message”（消息），在顺序图“基本场景”中，将“前台服务员”对象与“申请界面类”对象连接起来。双击该消息箭头，打开特性窗口，在“消息”输入框中输入“//录入路线代码和出发日”，表示消息内容：



点击“确定”即可。

“返回消息”的绘制：如图中消息 1.1.4，先用上述方法绘制该消息，然后在其特性窗口中，点选“控制流类型”区域中的“是返回”：



消息特性

签名

消息 (m): //旅游团和路线() 操作

参数 (a):

返回值 (v): void ☒ 显示继承的方法 (a)

分配给 (e):

构造型 (c):

别名 (l):

顺序表达式 (e)

条件 (d):

约束 (c):

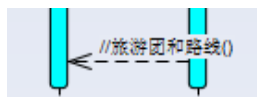
☐ 重复 (r)

控制流类型 (t):

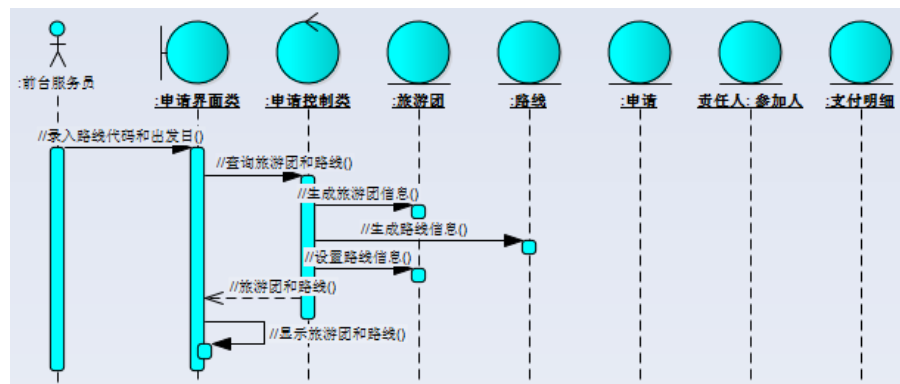
同步: 同步 生命周期:

类型: 调用 ☒ 是返回

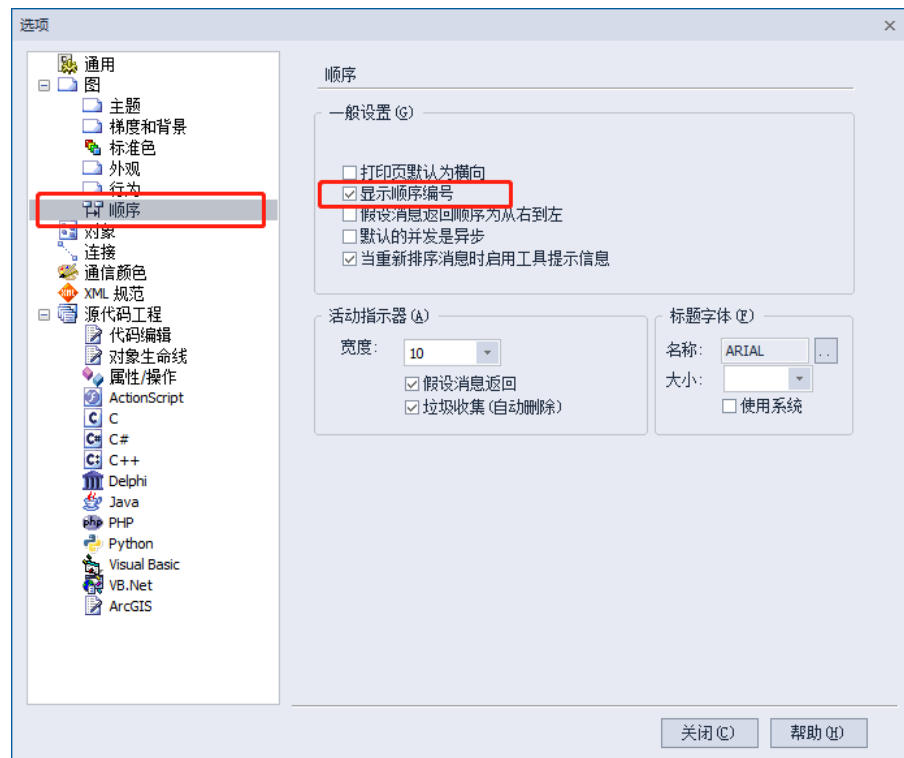
即可得到以虚线表示的“返回消息”：



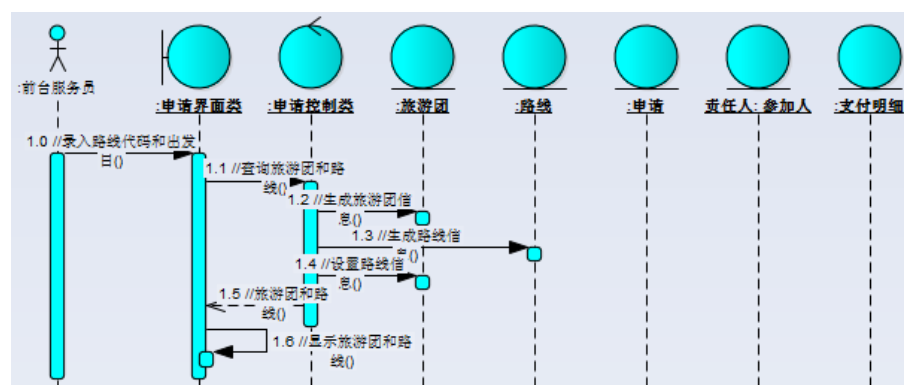
消息编号的显示：EA 工具中的顺序图默认不显示消息的编号，因为消息的顺序关系即为从上至下的关系。按上述绘制顺序图的步骤，得到的（部分）顺序图效果为：



注意到每条消息均无编号。如果希望显示编号，可以采用以下方式：在 EA 工具菜单栏中，选“工具-选项”，在打开的“选项”窗口中，从左边选择“图-顺序”标签，在右边“一般设置”里，在“显示顺序编号”处选中：

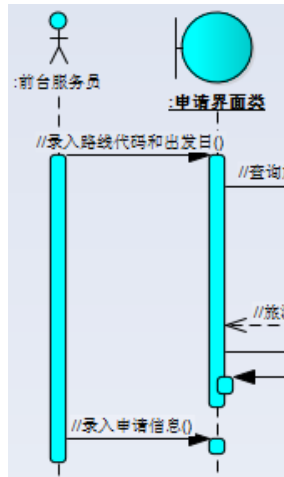


关闭该窗口后，设置立即生效：

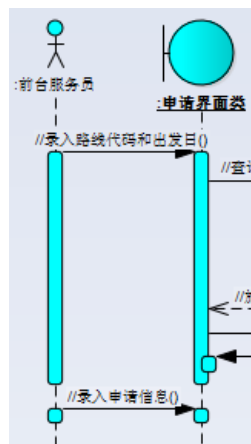


消息编号方式与图 5 稍有不同，没有关系。是否开启编号显示依据个人喜好，无明确要求。

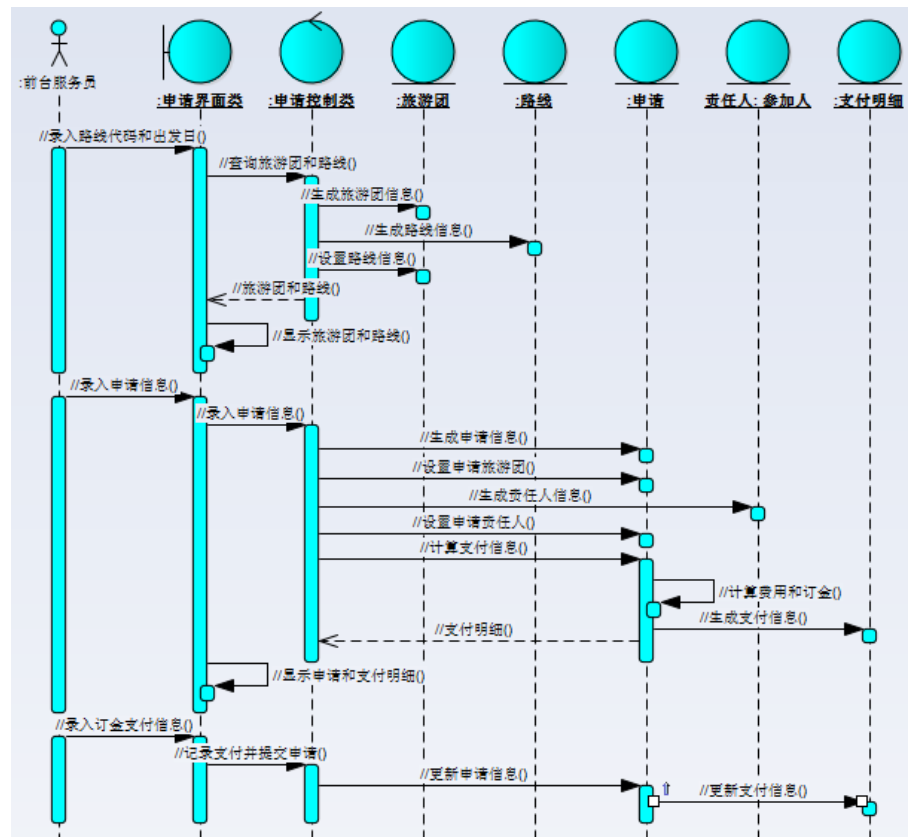
“执行”的分段：继续顺序图的绘制，我们发现在绘制“前台服务员”对象到“申请界面类”对象的消息“录入申请信息”时，该消息与“前台服务员”发出的上一条消息“录入路线代码和出发日”同属一段“执行”：



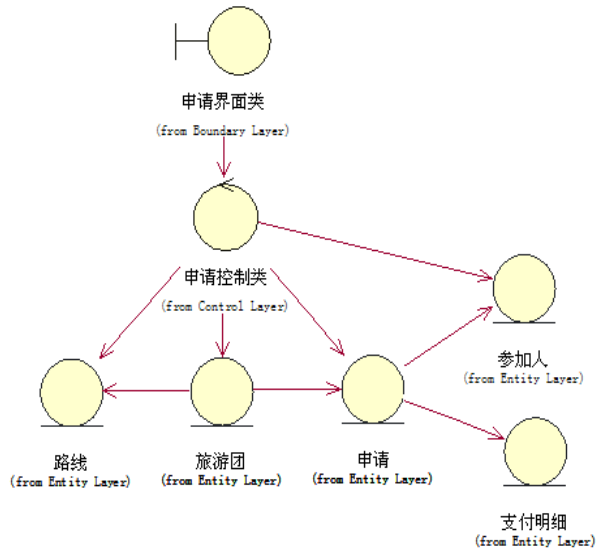
但从业务上理解，该消息开启了一段新的交互，与已进行的交互（由消息“录入路线代码和出发日”发起）相对独立，因此我们希望能图中表示出这种相对独立性，将该“执行”分段。右键选中消息“录入申请信息”，选择“激活-启动新消息组”，此时该消息使“前台服务员”的“执行”分段：



继续完成其它消息的绘制，效果图如下：



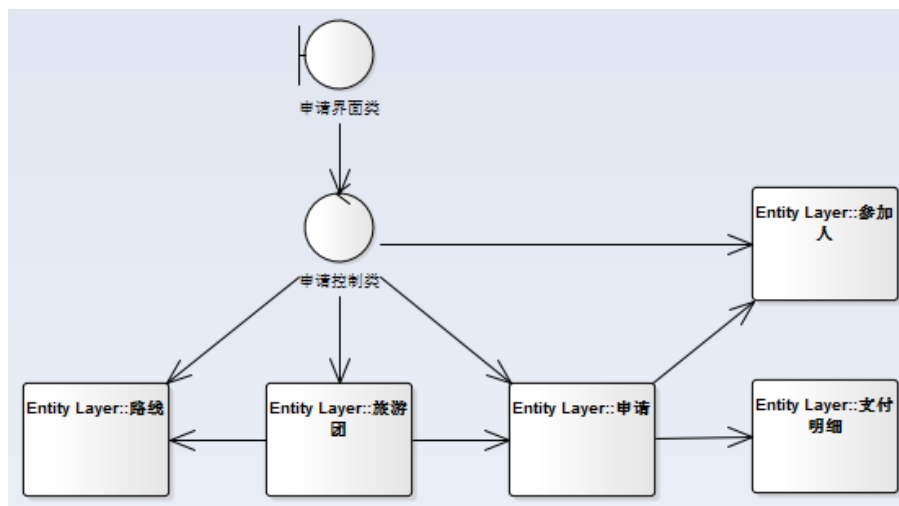
### 3.4 练习 4 构造用例实现-绘制 VOPC 类图



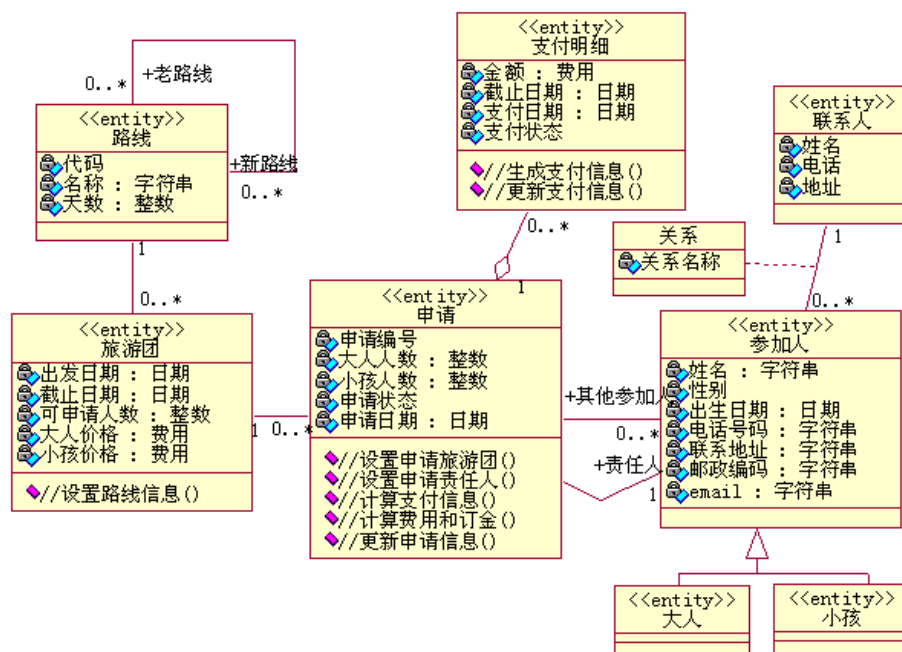
本练习的目的是为用例实现“办理申请手续-用例实现”绘制《06. 用例分析.pptx》第 64 页的 VOPC 类图（上图）。

首先在“项目浏览器”中，右键点击用例实现“办理申请手续-用例实现”，选择“增加-添加图”，图的名称命名为“VOPC”，类型为“类图”（UML Structural - Class）：

在“VOPC”图中拖入已在“Boundary Layer”、“Control Layer”、“Entity Layer”三个包中已定义的分析类，并使用“关联”关系(Associate)连接各个类，效果图如下：

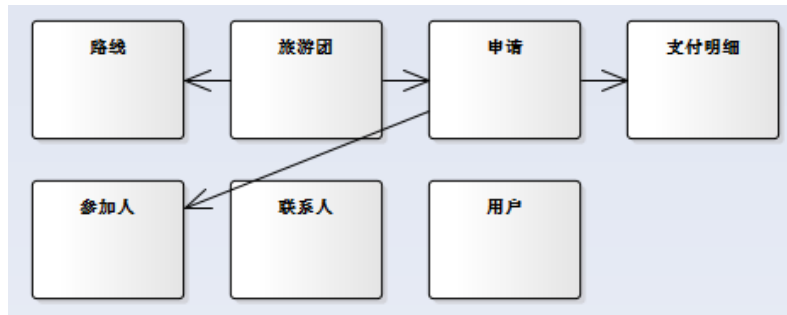


### 3.5 练习 5 完成实体类类图



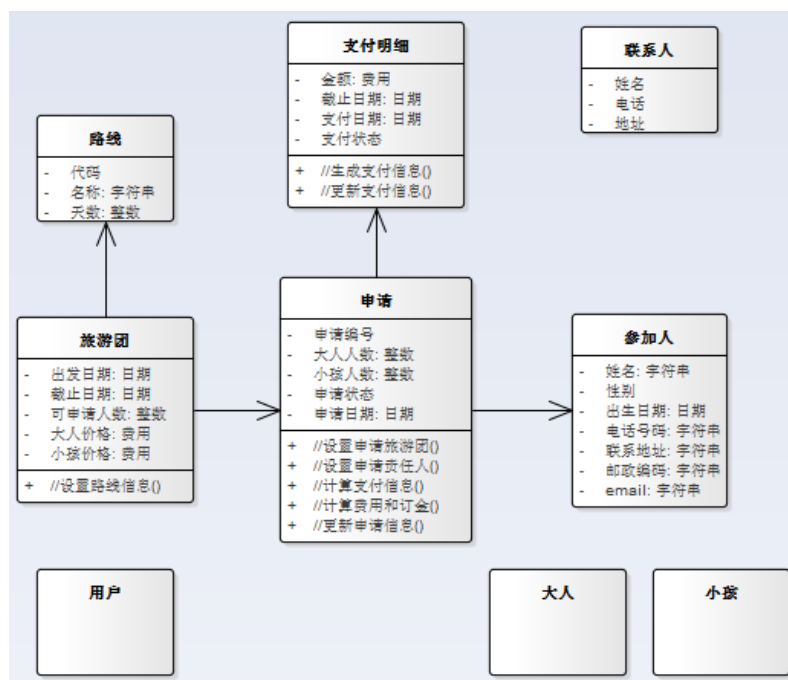
本练习的目的是绘制《旅游业务申请系统系统分析.pdf》中图 6 的实体类类图（上图）。

经过练习 4 中 VOPC 类图的绘制，我们发现“Entity Layer”的主视图中的实体类关系已随之发生改变：

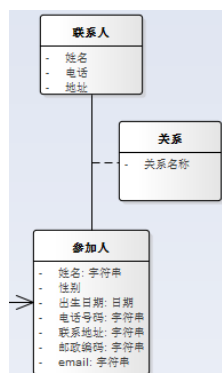


本练习的实体类类图，指的就是“Entity Layer”的主视图，我们在此图中对所有实体类及其之间的关系进行详细定义。

依照实验 1 中的方法，给各实体类增加属性和操作：

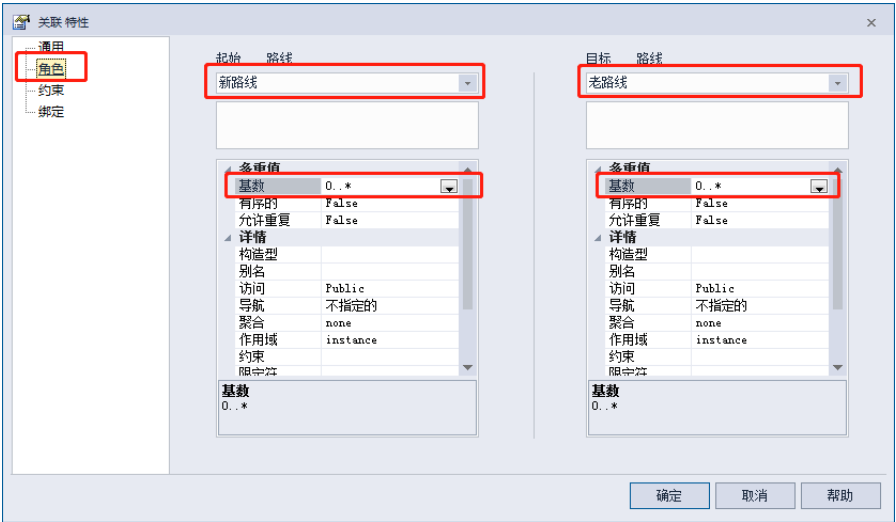


“关联类”的绘制：注意到“联系人”类与“参加人”类之间的关系用“关联类”来表示，在“工具箱”中选取类关系中的“Association Class”（关联类），将图中“联系人”类与“参加人”类进行连接，该关联类命名为“关系”，并添加相关属性：

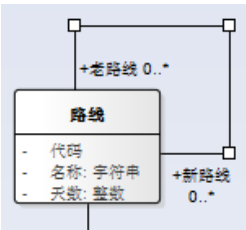


接下来对图中的类之间的关系进行修改完善。

关联关系端点名及多重性的指定：以“路线”类上的自反关联关系为例，双击该关联关系，在其特性窗口中选择“角色”标签，在右侧“起始”“目标”区域分别定义其角色名称以及“基数”：



点击“确定”即完成指定：



完成对图中所有关系的细化，最终效果图如下：

