

需求建模

需求是项目开发的基础，描述系统必须满足的条件或具备的能力，而 UML 用例建模技术则是一种非常有效的需求定义手段。通过本部分的学习能够了解需求的基础概念，并掌握利用用例技术进行需求建模的方法和实践过程，并能够动手完成某一给定系统的用例模型。

1 识别参与者

构建用例模型的第一步就是识别模型中的参与者。参与者代表了以某种方式与系统交互的人或事；它是在系统之外，透过系统边界与系统进行有意义交互的任何事物。

运用课堂教学中所提到的识别参与者技巧，从案例描述中获取目标系统的参与者。当然，在实际系统开发时，这个过程是在与用户沟通过程中完成的：通过与用户沟通，记录那些系统的各种外部因素（包括系统用户、外部系统和外部激励等），再从系统的职责入手分析这些外部因素，从而确定是否作为参与者，如果是参与者，则从系统角色的角度给出合适的名字。为了使该过程更具操作性，可以采用表 1 的方式来逐步分析和确定参与者。

表 1. 获取系统参与者

抽取角度	外部事物种类	主要日常工作	使用目标系统职责	参与者	典型代表
相关用户	前台招待顾客的员工	洽谈客户事宜并为客户办理各种申请和取消手续、完成费用支付	办理申请手续以及相关的取消、支付等后续业务	前台服务员	具体用户代表
	负责催款的员工	打印和邮寄旅游确认书和交款单	打印旅游确认书和交款单	收款员工	...
	旅行社内的会计人员	财务记账	不使用本系统，不是参与者		...
	宣传和路线管理员工	制作宣传资料、定期维护旅游路线和活动	维护旅游路线和旅游活动	路线管理员	...
其他外部事物	财务系统	记账等财务操作	接收本系统中与现金相关的财务信息	财务系统	...
	外部激励	关注或影响系统的运行	定期自动导出财务信息	时间	

从表中可以看出，该系统存在五个参与者，根据他们在系统中的职责给出了合适的名字。而其中旅行社内的会计人员并不能成为参与者，因为他并不直接使用该系统，而是使用财务系统进行日常工作。而财务系统与本系统存在信息交互，因此也作为一个参与者存在。另外，由于系统每天晚上需要定期运行，因此还需要定义一个时间参与者。获得了这些参与者之后，将这些参与者在建模工具中绘制出来，如图 1 所示。

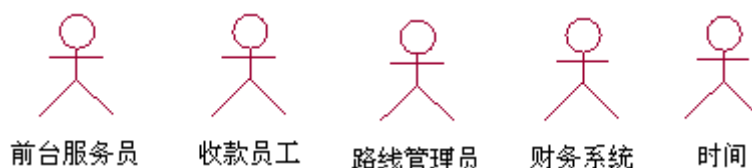


图 1 旅游申请系统参与者视图

2 识别用例

有了这些参与者之后，就可以从参与者使用系统的职责入手来定义用例；参与者使用系统所要达成的一个目标（或者说所要完成的一项工作）就作为一个用例而存在；当然如果这项工作不通过系统来完成就不作为用例存在。这个过程也可以使用表 2 的方式来完成

表 2 从参与者的角度获取用例

参与者	主要工作	是否使用系统	用例
前台服务员	向申请人介绍申请情况	否	
	为申请人办理申请手续	是	办理申请手续
	对申请参加人的增、删、改、查等日常维护	是	管理参加者
	记录申请人支付信息	是	完成支付
	为申请人取消申请	是	取消申请
收款员工	打印旅游确认书和余额交款单	是	打印旅游确认书和余额交款单
	邮寄旅游确认书和余额交款单	否	
路线管理员	制作宣传资料	否	
	设计旅游路线	是	管理路线
	设计旅游团（活动）	是	管理旅游团
	调整旅游团价格	是	设定价格
财务系统	记账等财务操作	否	
	接收与现金相关的财务信息	是	导出财务信息（被动）
时间	定期导出财务信息	是	导出财务信息
其它辅助用例	系统要区分各种不同的用户身份，并提供不同的功能	是	登录

这里主要存在的问题是“办理申请手续”用例的粒度问题，案例描述中有很长一段篇幅介绍了办理申请手续的过程，即员工首先要查询申请信息，之后录入申请责任人和其它相关信息，最后还要录入订金信息并打印收据。系统分析师有可能会把这里的几个步骤分成几个用例来表示，但仔细考虑前台服务员所完成的这些步骤实际上只有一个目的，即为申请人办理完成相关的申请手续，这些步骤是紧密联系在一起的。因此只需要定义一个“办理申请手续”用例即可描述用户目标，至于这些细节将在下一阶段作进一步描述。当然，其中的某些步骤可能可以独立存在，如“查询申请信息”，而且前台服务员也可能需要这样一个独立的业务，此时可以把这样独立存在的步骤独立形成一个子用例，这部分内容将会在重构用例模型中完成。但由于目前只是用例建模的早期，对于每个用例的内部细节还没有描述，因此现阶段也没有必要进行这样的分解。

3 构造用例图

识别这些参与者和用例之后，就可以采用用例图把它们表示出来，图 2 就是该系统的初始用例模型。注意参与者和用例之间的关联关系的定义，以及关联方向的含义。此外，在画用例图时注意图形的美观，采用合理的布局保持图形整洁、清晰，尽量不要出现一些线和图形的交叉等容易混淆的情况。

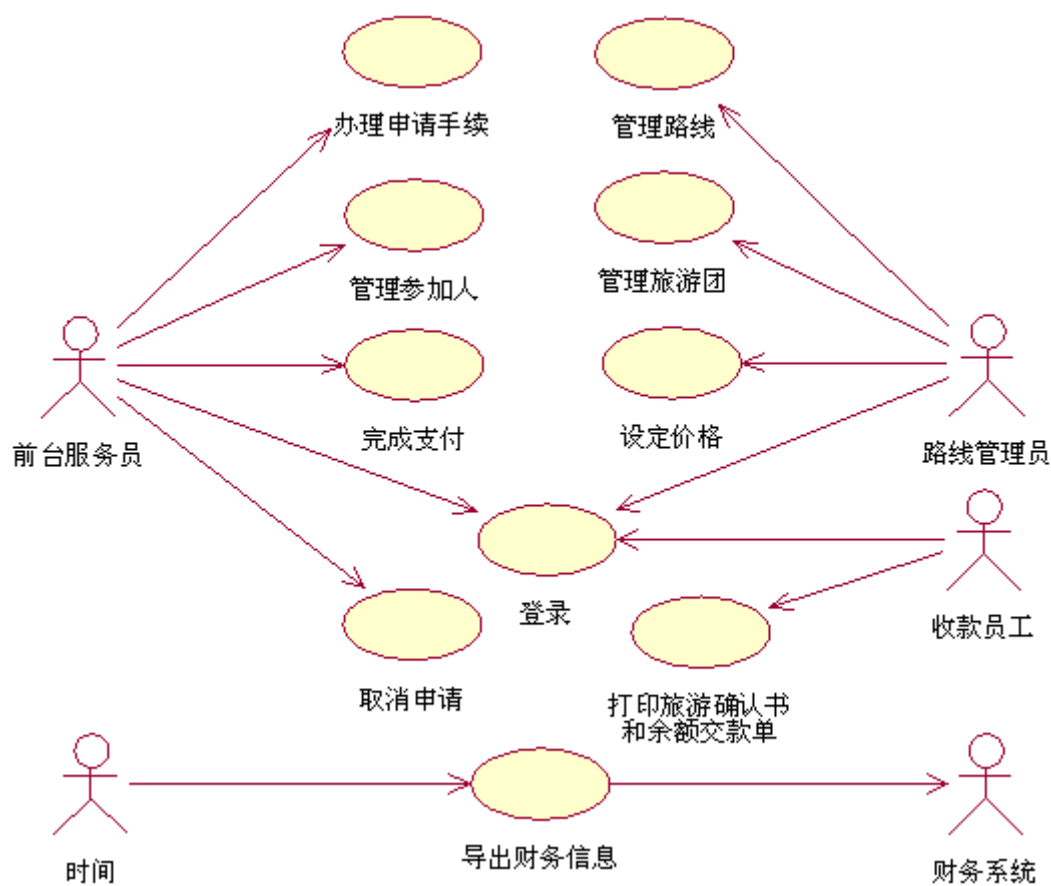


图 2. 旅游申请系统初始用例图

4 编写用例文档

绘制出系统的初始用例图后，整个用例建模过程其实才刚刚开始；更主要的工作就是要描述用例内部的处理细节，这是很多用例初学者容易忽略的问题。事实上，需求作为开发方和用户所达成的契约，必须要定义得非常具体而且准确；仅仅通过一幅图形显然是无法满足要求的。而且，正如前面所说，一个用例是用户所需要达到的目标，为了达到这个目标用户需要和系统进行频繁的交互，这些交互过程也无法在用例图中体现出来。为此，在完成用例图后，需要对图中的每一个用例进行详细的描述，这个描述的过程就是编写用例文档。通过文档的方式将用户与系统交互的过程一一记录下来，从而为以后的分析和设计提供基础。可以这样来描述用例图和用例文档的关系：用例图是整个需求的骨架，而用例文档则是需求的肉。也就是说，通过用例图建立了需求模型的基本结构，而需求的内容则需要通过用例文档来表示。此外，用例文档是与单个用例关联的，需要为每个用例编写一份独立的文档。

下面给出了旅游申请系统的两个核心用例的文档，注意其中一些细节的编写方法。表 3 给出了旅游申请系统中“办理申请手续”用例的文档。

表 3 “办理申请手续”用例文档

用例名	办理申请手续
简要描述	前台服务员通过该用例为申请人办理申请旅游团的手续
参与者	前台服务员
涉众	申请人、前台服务员

相关用例	暂无								
前置条件	前台服务员登录到系统								
后置条件	申请信息被正确保存，相关旅游团可申请人数减少								
基本事件流 <ol style="list-style-type: none"> 1. 该用例起始于旅客需要办理申请手续； 2. 前台服务员录入要申请的旅游团旅行路线代码和出发日期； 3. 系统查询要申请的旅游团信息（A-1）； 4. 系统显示查询到的旅游团和相关路线信息（D-1）（A-2、A-3）； 5. 前台服务员录入本次申请信息（D-2）； 6. 系统显示旅行费用的总额和申请订金金额； 7. 前台服务员提交该申请信息； 8. 系统保存该申请信息（A-4），用例结束。 									
备选事件流 <p>A-* 前台服务员在提交该申请前，随时都可能中止该申请</p> <ol style="list-style-type: none"> 1. 系统显示中止确认的消息； 2. 前台服务员可以结束该用例，也可以选择继续录入下一个申请。 <p>A-1 无法查询到所需的旅游团信息</p> <ol style="list-style-type: none"> 1. 系统显示录入的旅游路线代码或者出发日期有误信息； 2. 前台服务员再次录入旅游路线代码和出发日期，也可以结束用例。 <p>A-2 旅行已超过申请截止日期</p> <ol style="list-style-type: none"> 1. 系统提示已超过申请截止日期，不能申请； 2. 前台服务员重新输入旅游路线代码和新的出发日期，也可以结束用例。 <p>A-3 可以申请的人数为 0 人</p> <ol style="list-style-type: none"> 1. 系统提示旅游团人数已满； 2. 前台服务员重新输入新的旅游路线代码和出发日期，也可以结束用例。 <p>A-4 保存信息失败</p> <ol style="list-style-type: none"> 1. 系统显示保存失败，并提示用户需要再次提交； 2. 前台服务员可以重新提交该申请，也可以结束用例。 									
补充约束-数据需求 <p>D-1 显示的旅游团和路线信息包括：旅游路线代码、旅游路线名称、出发日期、天数、申请截止日期、可申请人数、大人的单价和小孩的单价等。</p> <p>D-2 录入申请信息包括：申请责任人的姓名、电话号码、参加的大人人数、小孩人数</p>									
补充约束-业务规则 <p>B-1 所申请旅游团的截止日期在申请日期之前</p> <p>B-2 所申请旅游团的人数限额未满</p> <p>B-3 申请订金的计算规则如下表所示：</p> <table border="1"> <thead> <tr> <th>距出发日期的天数</th><th>订金比例</th></tr> </thead> <tbody> <tr> <td>≥2 个月</td><td>10%</td></tr> <tr> <td>≥1 个月，且<2 个月</td><td>20%</td></tr> <tr> <td><1 个月</td><td>全款</td></tr> </tbody> </table>		距出发日期的天数	订金比例	≥2 个月	10%	≥1 个月，且<2 个月	20%	<1 个月	全款
距出发日期的天数	订金比例								
≥2 个月	10%								
≥1 个月，且<2 个月	20%								
<1 个月	全款								
待解决问题 （暂无）									
相关图 （暂无）									

表 4 给出了“管理参加人”用例文档，与前一个用例文档不同的是，该用例文档的基本事件流被划分成三个子流程，通过子流程的方式可以使该用例文档的结构更加清晰。

表 4 “管理参加人”用例文档

用例名	管理参加人
简要描述	前台服务员通过该用例对申请参加人的信息进行维护
参与者	前台服务员
涉众	申请人、申请参加人
相关用例	暂无
前置条件	前台服务员登录到系统
后置条件	申请参加人的信息被正确的录入到系统中
<p>基本事件流</p> <ol style="list-style-type: none"> 用例起始于前台服务员需要对申请的参加人信息进行维护； 前台服务员输入查询条件（D-1），查询申请信息； 系统查询该申请（A-1），并显示申请详细信息（D-2）； 前台服务员选择所要进行的操作； 系统根据前台服务员选择的操作，执行以下的子流程： <ul style="list-style-type: none"> 选择“增加参加人”操作时，开始“增加参加人”子流程（S-1）； 选择“修改参加人”操作时，开始“修改参加人”子流程（S-2）； 选择“删除参加人”操作时，开始“删除参加人”子流程（S-3）； 子流程完成后，用例结束。 <p>子流程 S-1：增加参加人</p> <ol style="list-style-type: none"> 系统显示申请责任人的姓名和电话号码； 前台服务员录入申请责任人信息（D-3）； 前台服务员录入申请责任人旅行途中的联络人信息（D-4）； 前台服务员继续录入其它参加人的信息； 前台服务员录入参加人信息（D-3）； 前台服务员录入参加人有关旅行途中的联络人信息（D-4）； 重复步骤 5、6，录入所有的参加人（A-2）； 前台服务员提交本次录入信息（A-3）； 系统保存参加者信息（A-4），结束该子流程。 <p>子流程 S-2：修改参加人</p> <ol style="list-style-type: none"> 系统显示全部参加人的姓名； 前台服务员选出要修改的参加人； 系统显示要变更的参加者信息（D-3）和联络人信息（D-4）； 前台服务员修改相关的信息； 前台服务员提交本次修改（A-2）； 系统保存参加人信息，结束该子流程。 <p>子流程 S-3：删除参加人</p> <ol style="list-style-type: none"> 系统显示全部参加人的姓名； 前台服务员选出删除的参加人； 系统显示取消手续费用和返还金额； 前台服务员确认删除； 系统保存本次删除信息； 若删除的参加人就是申请责任人，为了选择新的申请责任人，系统会显示所有参加 	

人的姓名； 7. 前台服务员选择新的申请责任人； 8. 系统录入新的申请责任人（A-4），结束该子流程。										
备选事件流 A-* 前台服务员在操作提交之前，随时都能够结束子流程 1. 系统显示确认中止的消息； 2. 前台服务员可以结束子流程，也可选择继续其它操作。 A-1 没有找到申请信息 1. 系统提示未找到该申请信息 2. 前台服务员输入查询条件进行查询，也可以结束用例。 A-2 必填项有未输入项目 1. 系统提示有未输入项目； 2. 前台服务员再次输入未输入项。 A-3 尚未录入所有参加者的信息 1. 系统提示有未录入的参加者信息； 2. 前台服务员可以继续录入参加者的信息，也可登录目前已录入的参加者的信息，结束子流程。 A-4 系统保存失败 1. 系统提示保存失败； 2. 前台服务员可以再次提交，也可结束该用例										
补充约束-数据需求 D-1 查询条件包括：旅游线路代码、出发日期、申请责任人姓名 D-2 显示的申请信息包括：旅游线路代码、旅游团名称、出发日期、申请日期、申请责任人姓名、支付情况。 D-3 参加人信息包括：性别、出生年月、现在住所、邮政编码、E-Mail 地址等 D-4 联络人信息包括：姓名、与本人关系、住址、邮政编码、电话号码等； 补充约束-业务规则 B-1 取消手续费如下表所示： <table><tr><td>距出发日期的天数</td><td>取消手续费</td></tr><tr><td>1 个月以上</td><td>无</td></tr><tr><td>1 个月到 10 天</td><td>20%</td></tr><tr><td>10 天到 1 天（前一天）</td><td>50%</td></tr><tr><td>0 天（出发当天）</td><td>全款</td></tr></table>	距出发日期的天数	取消手续费	1 个月以上	无	1 个月到 10 天	20%	10 天到 1 天（前一天）	50%	0 天（出发当天）	全款
距出发日期的天数	取消手续费									
1 个月以上	无									
1 个月到 10 天	20%									
10 天到 1 天（前一天）	50%									
0 天（出发当天）	全款									
待解决问题 （暂无）										
相关图 （暂无）										

2.5 重构用例模型

有了基本的用例模型和相应的用例文档，就完全可以对各类应用的需求进行定义。然而，对于大规模的复杂应用而言，仅使用这些可能还是不够的。在这些大规模系统中，可能会存在十几个甚至几十个用例，过多的用例也使得用例图变得过于复杂；而且每个用例的规模也存在很大的差别，

有的比较简单（如登录），有的特别复杂，其文档可能就有十几页甚至几十页；此外，每个用例的重要程度也不一样。针对这些问题，需要采用更多的用例建模技巧，来对初始的用例模型进行重构，以便构造规模适中、结构合理的用例视图。在本案例中主要采用用例关系来重构用例模型。通过用例关系将复杂的用例进行适当的分解，以便于提高需求的复用性和可扩展性等，从而使用例模型的结构更合理。

在编写旅游申请系统的用例文档时，可以发现“管理参加人”和“完成支付”中都首先需要查询已有的申请信息，并显示申请的情况；虽然有些细节的差异，如管理参加人时并不需要显示有关余款的信息，而完成支付时还可以按照交款单编号来查询。但这些细微的差别可以通过适当地调整用例文档来解决（如管理参加人虽然不需要支付信息，但查询到这些信息也没什么影响，只是不需要处理而已）。因此，从这两个用例中就可以抽取出公共行为“查询申请信息”，而这个公共行为即可构成一个单独的被包含用例“查询申请信息”。这三个用例之间的关系如图3所示。注意图中包含关系的方向，由基用例指向被包含用例。

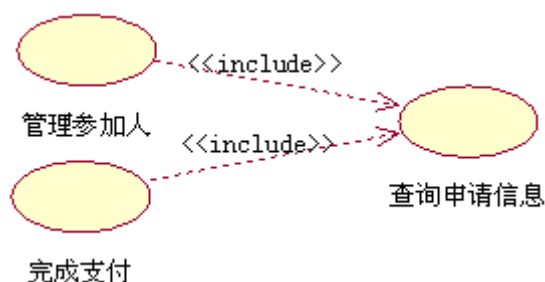


图3 用例间的包含关系

通过包含关系对用例模型进行重构后，相应的用例文档也会受到影响。原有的用例中已经抽取出了的部分就不再出现在用例文档中，而是通过引用被包含用例来表示。当然，还需要为新增加的“查询申请信息”用例编写相应的用例文档。

在编写旅游申请系统的用例文档时，可以发现“导出财务信息”用例中存在一些复杂的异常处理行为（即 A-2 备选流），这部分行为在当前用例中并没有展开论述；而且这些异常处理方式本身和当前导出财务信息的业务没有太大的关系。因此，完全可以把此部分内容分离出去，构成扩展用例，修改后的用例图如图4所示。

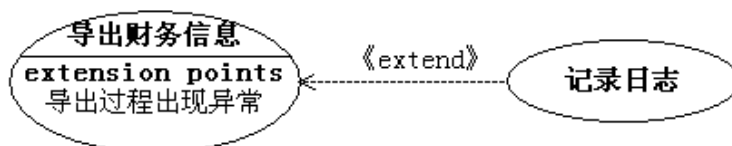


图4 用例间的扩展关系

从图中可以清楚地看出基用例“导出财务信息”定义了一个扩展点“导出过程出现异常”，而扩展用例“记录日志”则是基于该用例进行扩展的。需要说明的是，目前很多工具并不将扩展点在用例图中显示出来（EA 工具可以支持），但不管是否显示出来，只要存在扩展关系，基用例中就存在相应的扩展点；而且有关扩展点和扩展关系也还需要在用例文档中体现。