

论面向对象的软件设计

——UML 在面向对象软件架构中的应用

摘要

自“软件危机”出现过后，工程化软件开发方法不断发展，采用什么方法对大规模软件进行设计并保证软件的质量。在这样背景下，人们开始从面向数据流过程开发法中不断思考，进而引入对象的概念。对象是数据与行为的封装，对象既是自然界中的对象，这种方法不仅易理解也符合事物本身结构，对象之间通过消息进行交互。面向对象的分析与设计不断发展，UML 成了面向对象分析与设计的形式化表示方法。本人在 2016 年，采用面向对象的方法，对 XX 航空公司的航路费及机场费核算[RAFS]系统进行了分析与设计，并取得了成功，通过本项目让我认识到面向对象软件分析与设计的应用场景，面向对象分析与设计的过程，方法。并且加深了对面向对象理念的理解，如用例在需求阶段的重要作用，领域概念模型构建的方法，类识别方法等。

正文

1、面向对象分析

面向对象分析方法的核心是用例和用例图。用例是用户与系统交互的功能集合的说明，这里用户是一种角色，可以是其他系统，也可以是第三方系统接口。用例核心是一种功能分解方法，主要用于捕获软件需求。用例的核心是用例描述，用例描述中的主要内容有，用例名，用例编号，用例角色，主流程，异常流程，错误处理，用途描述及特征，前置条件，后置条件等。

1.1 确定用例

通过对使用 RAFS 系统用户及交互系统的调研与分析，RAFS 系统包括以下用例：

用户认证与授权，机场费航路费起降规则编辑，机场费航路费计算，报表与查询等模块。

其中核心用例为机场费航路费规则编辑，机场费航路费计算。费用规则编辑的复杂性存在于，每个国家的机场费航路费的收费规则相差很大。首先计费规则复杂。比如，有的机场跟据不同时间段，飞机的不同起飞重量进行计费；也有根据不同的时间段以不同费率收取噪音费；也有是固定费率进行计算；航路费指航线所经过的国家领空，这个国家收取的费用。费用规则一般是按里程固定费率计算。其次，一个费用有多个计费规则，有“并”关系也有“或”关系。因此决定规则编辑界面规则的设计变得困难。

1.2 复杂用例的流程描述

活动图主要用于描述对象的操作流程与事件处理，状态图用于描述对象的状态。活动图与状态图可以用于描述用例的动态行为及用例中核心对象的状态转换。在 RAFS 系统中，通过对各国机场费，航路费收费规则文档的分析，发现规则可以分为条件和结果。条件则相对固定几类：1，算术运算条件；2，固定费用选择类；3，单项条件类，4 多选条件类。这正符合界面组件的面向对象设计。条件组件是一个界面组件，有自己的生命周期，因此对条件对象绘制了状态图，确定条件对象有初始状态，展现状态，编辑状态，结束状态。条件对象有始化，装入数据，渲染，数据更新，重绘界面，提交数据，销毁等过程。针对条件中的飞行数据则来自于第三方系统，为了便于在规则界面组装成条件，架构组对这些数据进行了分析，将核心字段提取出来形成数据因子，用于构成条件规则。

费用计算用例主要是根据配置的规则，结合实际的飞机飞行数据，算出费用。难点在于根据规则中配置的因子，找到因子对应的实际飞行数据，将数据应用于规则，算出实际费用。

1.3 构建用例图根据识别到的用例，架构组构建了用例图，用例进一步细化出了如

下用例。费用编辑用例细化出了，配置因子，配置费用规则。费用计算用例又细分出，航路费计算，起降费计算。用户认证授权细分出，用户身分识别，用户权限管理用例。进过与客户多次沟通，最终确定了用例图，用例描述，核心用例的流程，核心对象的状态，及主要业务对象之间的业务操作流程。

1.4 构建领域模型

识别出用例后，根据用例，架构组将用例中的使用到的名词，全部写到卡片上，然后将卡片和用例发给架构组成员，让成员确定最重要的名词，和不需要名词。最后进行架构组会议，确定了高层的领域对象。确定高层次的领域对象后，架构组对每个对象进行属性提取。这一步操作过后，是确定领域对象之间的关系。RAFS 系统初步识别出如下对象：因子，条件，表达式，规则，结果，费用，飞机型号，国家，城市，飞机制造商等对象。

1.5 顶层架构图

通过对用例，领域对象的分类，形成 UML 包图，构建出粗粒度的架构图，确定各模块之间的关系。为接下来的工作打好基础。根据客户的非功能性需求，此系统必须用浏览器进行访问。因此架构组认为除费用计算用例以外的用例整合为一个基于 B/S 架构风格的系统，采用 MVC 为核心处理模式。MVC，主要是将展现与控制进行有效分工，模型保存业务数据。模型的变化会通知视图更改界面，视图通过控制器处理业务或者调用后端业务层接口，根据业务需要选择不同的视图，将视图展现给用户。

经过以上的分析与处理，最终确定了系统的功有需求列表，确定了需求范围。形成了需求定义说明书，其中包含了所有用例，用例图。每个用例使用单独的用例说明文档。

2，面向对象设计通过 OOA 过程后，系统进入设计阶段。设计阶段主要包括以下过程。实现用例方案；设计技术支撑方案；设计用户界；精化设计。本节只对各个过程中

的核心子过程进行描述。

2.1 类识别

根据需求分析文档、用例定义文档及初步的领域模型，提取边界类，控制类，实体类，辅助类。这一过程，RAFS 系统识别出以下界面组件，规则编辑界面类，规则编辑界面类，规则条件界面类（包含算术条件类，单选条件类，复选条件类，固定费用条件类），用户登录界面类，计算进度显示等界面类。控制类：界面组件控制基类；4 个条件控制类等其他控制类。实体类主要是精化领域模型中抽象出的模型类。另外还抽取规则业务处理辅助类，因子业务处理辅助类，数据访问等相关类。同时还对相类的关系进行的标识，如规则类是条件类的聚合，条件组件抽象类是 4 个条件类的泛化等等。

2.2 构造交互图

2.3 技术选型

通过以上几个阶段，架构组对 .net 平台，java 平台进行了评估。就以下几个方面进行讨论与权衡：

- 1.平台成熟度，及大型项目中的成功案例
- 2.平台的基础服务的成熟度
- 3.开发人员招聘的难易成度
- 4.与企业现有系统的技术的互操作程度

最终选择 Java 平台作为项目开发的基础平台。在前端规则界面数据的存储与传输中采用 XML 格式。界面组件采用 ZK8.0。在 JDK1.7, 版本的基础上使用 Spring MVC 为表现层，Spring 作为轻量级的业务层集成框架，使用 hibernate，O/R mapping 的方式进行数据访问。费用计算采用成熟的规则引擎 Drools，采用模板引擎将规则文件（XML）转成 Drools 的规则文件。因子数据的存储采用星型结构提高因子数据的访

问效率。为了便于后期的扩展，本系统中所有功能以接口的方式进行定义，真正做到对修改关闭。同时在模块设计中制定了上层仅依赖下层原则保证模块的独立性。

3，架构文档化

经过需分析与设计最终确定的开发中使用的技术，明白了目标系统概念模型，功能模型。形成了以类图，用例图，活动图，交互图，状态图，构件图，部署图为中心的UML架构文档。并结合4+1视图对整体架构进行了描述。最终形成了架构设计说明书，架构质量规格说明书，并交付客户。

结论

通过本项目，本人进一步理解了面向对象的需求分析和设计过程，对UML的静态模型图，动态模型图的运用有了新的体会。通过本项目我认为面向对象的分析与设计，核心是抓信用用例图，然后针对用例的描述对静态模型，动态模型进行细化。最终根据设计模型的特点选择实现技术。并且这个过程不是一次性的而是迭代多次，不断精确的过程。最终根据选择的技术特点及系统设计，业务要求采用合适的开发方式。

 <p>免费手机刷题</p>	 <p>免费电脑刷题 网址: t.51cto.com</p>	 <p>公众号: 51CTO 软考</p>	 <p>微信交流社群 更多备考信息及福利</p>
--	---	--	--

51CTO 软考