4.1.1 orderbl模块

(1)模块概述

orderbl模块承担的需求参见需求规格说明文档功能需求和相关非功能需求。

orderbl模块的职责及接口参见软件系统结构描述文档表10。

(2)整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。比如展示层和业务逻辑层之间，我们添加businesslogicservice.orderblservice.Orderblservice接口。业务逻辑层和数据层之间添加dataservice.orderdataservice.Orderdataservice接口。OrderPO是作为订单记录的持久化数据被添加到设计模型中去的。

orderbl模块的设计如图15所示。

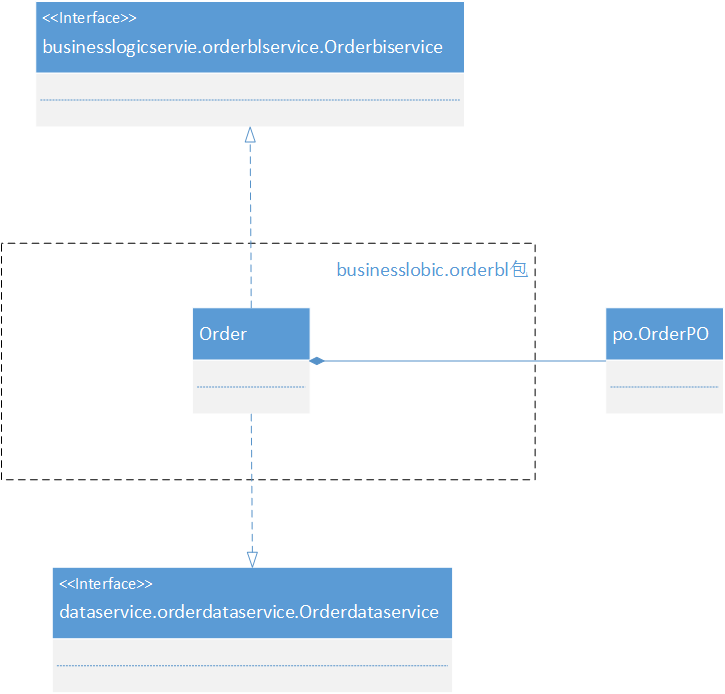


图15 orderbl模块各个类的设计

orderbl模块各个类的职责如表所示。

表 orderbl模块各个类的职责

|  |  |
| --- | --- |
| 模块 | 职责 |
| Login | 系统用户的领域模型对象，拥有用户数据的领域模型对象，拥有用户数据的编号密码，可以解决登陆问题 |
| Order | 有关订单的领域对象模型，可以帮助完成订单界面所需要的服务 |

(3)模块内部类的接口规范

Order的接口规范如表所示。

表 Order的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| Order.add | 语法 | public OrderVO add(String sender, String addressee, String desti, String startingPlace,  ExpressType eType, String name, String id, int num) |
| 前置条件 | 启动一个订单回合 |
| 后置条件 | 在一个订单回合中，输入寄件单信息生成订单 |
| Order.delete | 语法 | public void delete(String id) |
| 前置条件 | 展开一个订单列表 |
| 后置条件 | 根据id删除一个订单 |
| Order.modify | 语法 | public OrderVO modify(String sender, String addressee, String desti, String startingPlace,  ExpressType eType, String name, String id, int num) |
| 前置条件 | 选择一个订单 |
| 后置条件 | 修改一个订单的内容 |
| Order.inquireA | 语法 | public OrderVO inquireA(String id) |
| 前置条件 | 输入一个订单编号 |
| 后置条件 | 查看一个订单信息 |
| Order.inquireB | 语法 | public OrderVO inquireB(String date) |
| 前置条件 | 输入一个日期 |
| 后置条件 | 查看一个订单信息 |
| Order.generateId | 语法 | public void generateId |
| 前置条件 | 完成订单信息输入 |
| 后置条件 | 生成一个订单条形码号 |
| Order.getTotalPrice | 语法 | public double getTatalPrice(String startingplace,String desti,ExpressType sType,int num) |
| 前置条件 | 完成订单信息输入 |
| 后置条件 | 根据订单信息计算总费用 |
| Order.endOrder | 语法 | public void endOrder() |
| 前置条件 | 已结束订单处理 |
| 后置条件 | 结束本次订单处理，持久化更新设计的领域对象数据 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| Orderdataservice.add(OrderPO po) | 增加单一持久化对象 |
| Orderdataservice.delete(OrderPO po) | 删除单一持久化对象 |
| Orderdataservice.modify(OrderPO po) | 修改单一持久化对象 |
| Orderdataservice.inquireA(String id) | 根据id进行查找单一持久化对象 |
| Orderdataservice.inquireB(String date) | 根据日期查找多个持久化对象 |
| Orderdataservice.update(OrderPO po) | 更新单一持久化对象 |

(4)业务逻辑层的动态模型

下图表示了快递物流系统中，当快递员输入增加订单所需数据后，订单业务逻辑处理的相关对象之间的协作。

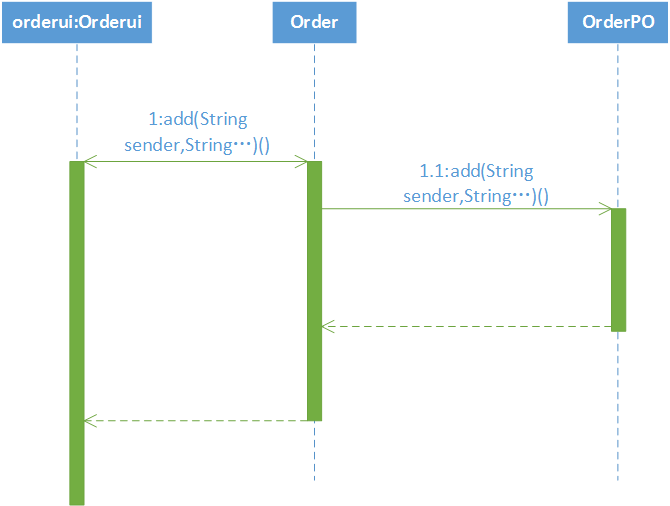


图 添加订单的顺序图

下图所示的状态图描述了Order对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着add方法的调用，调用generateId和getToalPrice方法，进入Order状态。其他方法也类似进行。

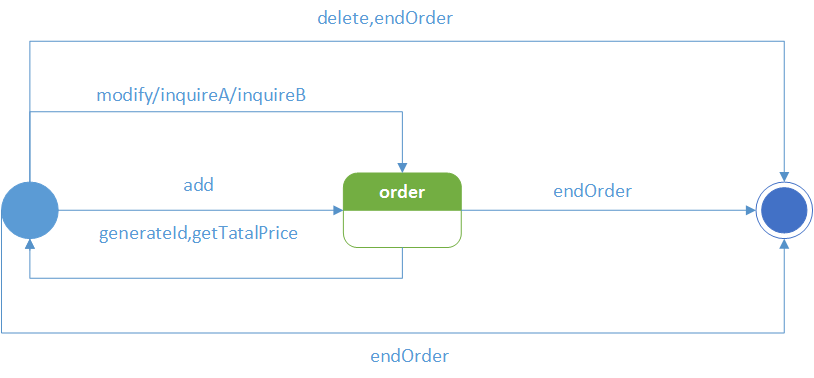


图 Order对象状态图

4.1.2 mngerandaccountantbl模块

(1)模块概述

mngerandaccountantbl模块承担的需求参见需求规格说明文档功能需求和相关非功能需求。

mngerandaccountantbl模块的职责及接口参见软件系统结构描述文档表10。

(2)整体结构

根据体系结构的设计，我们将系统分为展示层、业务逻辑层、数据层。每一层之间为了增加灵活性，我们会添加接口。mngerandaccountantbl模块包含三个部分：ApproDocm，InquireOperaLog，StatisAnaly，对应的，业务层和展示层之间，我们添加ApproDocmblservice,OperaLogblservice,StatisAnalyblservice接口；业务层和数据层之间，我们添加ApproDocmdataservice,OperaLogdataservice,StatisAnalydataservice接口。OperaLogPO是作为操作日志的持久化对象被添加到设计模型中去的。

mngerandaccountantbl模块的设计如下图所示。

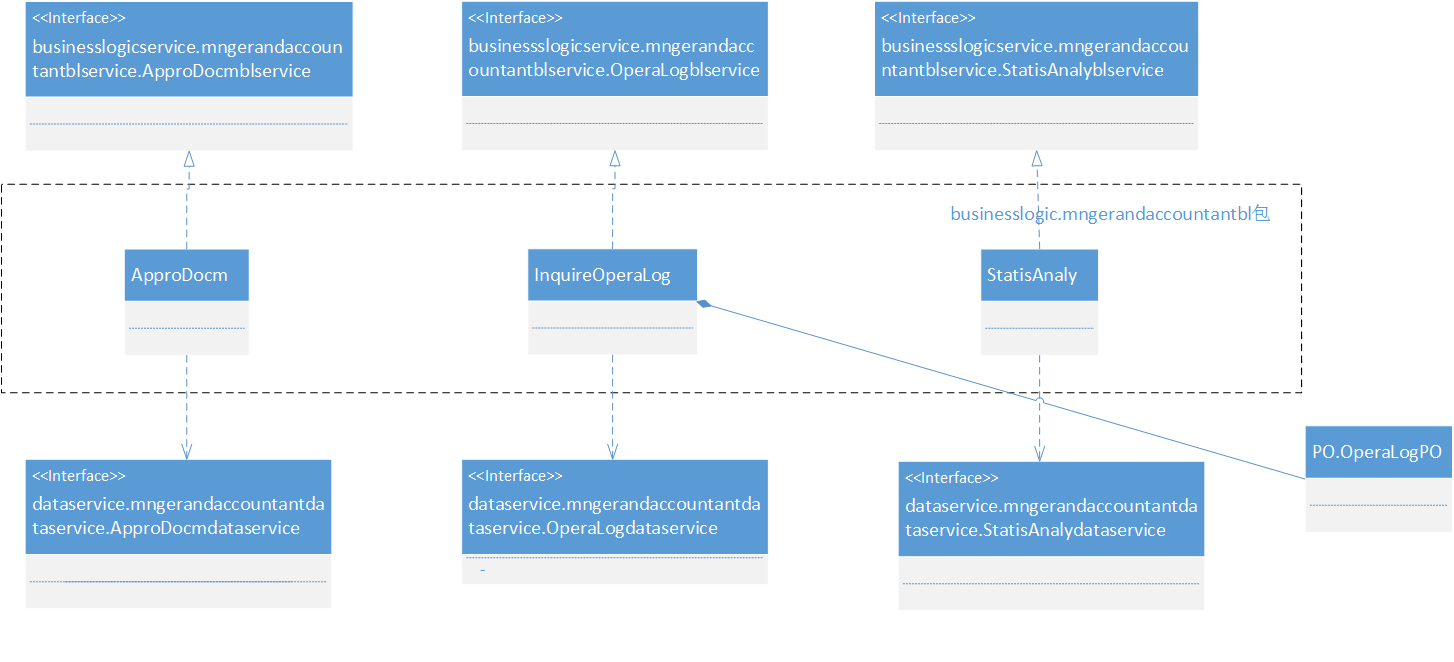


图 mngerandaccountantbl模块各个类的设计

mngerandaccountantbl模块各个类的职责如下表所示。

|  |  |
| --- | --- |
| 模块 | 职责 |
| Login | 系统用户的领域模型对象，拥有用户数据的领域模型对象，拥有用户数据的编号密码，可以解决登陆问题 |
| ApproDocm | 有关审批单据的领域对象模型，可以帮助完成审批单据界面所需要的服务 |
| InquireOperaLog | 有关查询操作日志的领域对象模型，可以帮助完成操作日志界面所需要的服务 |
| StatisAnaly | 有关统计分析的领域对象模型，可以帮助完成统计分析界面所需要的服务 |

(3)模块内部类的接口规范（以ApproDocm的接口规范为例）

ApproDocm的接口规范如表所示。

表 ApproDocm的接口规范

|  |  |  |
| --- | --- | --- |
| 提供的服务（供接口） | | |
| ApproDocm.chooseKind | 语法 | public ParentLineItemVO chooseKind(String kind) |
| 前置条件 | 启动一个审批单据回合 |
| 后置条件 | 选择一种单据，显示其列表 |
| ApproDocm.Inquire | 语法 | public ParentVO inquire(String id) |
| 前置条件 | 展开一个单据列表 |
| 后置条件 | 根据id查询一个单据的内容 |
| ApproDocm.modify | 语法 | public ParentVO modify() |
| 前置条件 | 选择一个单据 |
| 后置条件 | 修改一个单据的内容 |
| ApproDocm.delete | 语法 | public void delete(String id) |
| 前置条件 | 输入一个单据编号 |
| 后置条件 | 删除一个单据信息 |
| ApproDocm.finishApprove | 语法 | public void finishApprove(String id) |
| 前置条件 | 已审批 |
| 后置条件 | 标记一个单据已审批 |
| ApproDocm.mark | 语法 | public void mark() |
| 前置条件 | 已审批 |
| 后置条件 | 标记多个单据已审批 |
| ApproDocm.endAD | 语法 | public void endAD() |
| 前置条件 | 完成审批单据 |
| 后置条件 | 结束本次审批单据处理，持久化更新设计的领域对象数据 |

|  |  |
| --- | --- |
| 需要的服务（需接口） | |
| 服务名 | 服务 |
| ApproDocmdataservice.inquire(ParentPO po) | 查询单一持久化对象 |
| ApproDocmdataservice.modify(ParentPO po) | 修改单一持久化对象 |
| ApproDocmdataservice.update(ParentPO po) | 更新单一持久化对象 |

(4)业务逻辑层的动态模型（以ApproDocm为例）

下图表示了快递物流系统中，当总经理选择审批单据后，审批单据业务逻辑处理的相关对象之间的协作。

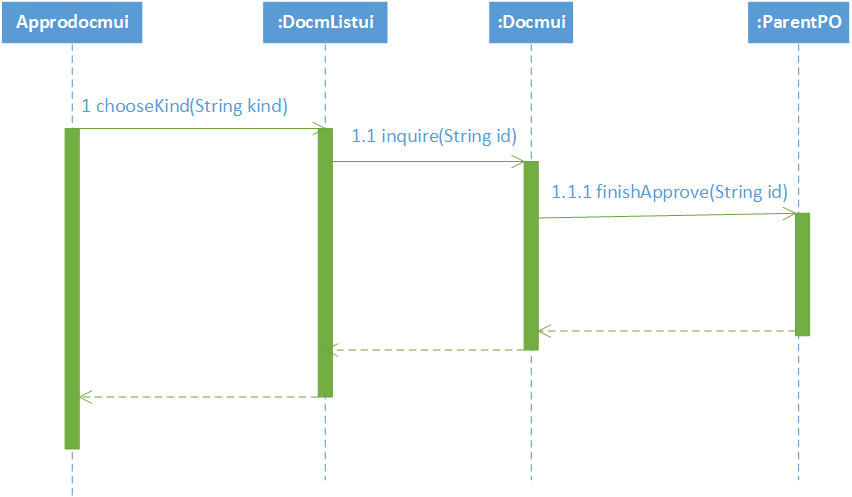


图 审批单据的顺序图

如下图所示的状态图描述了ApproDocm对象的生存期间的状态序列、引起转移的事件，以及因状态转移而伴随的动作。随着chooseKind方法被ui调用，进入list状态；经过inquire方法进入docm状态，最后通过finishApprove/mark方法确认为已审批状态。

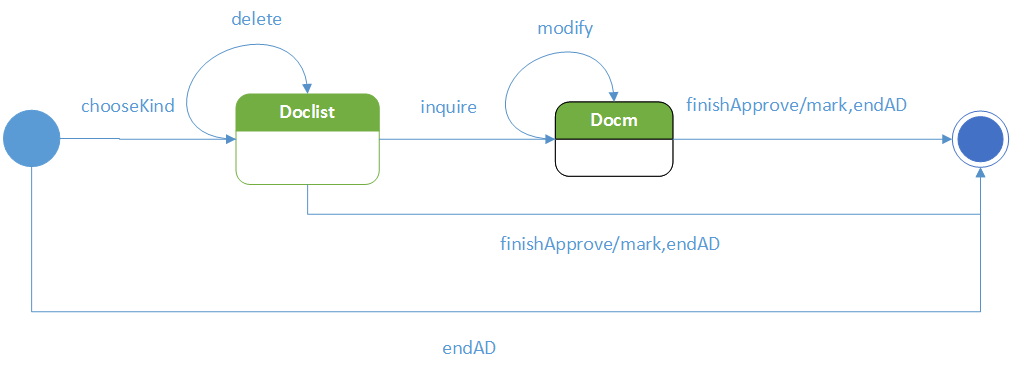


图 ApproDocm的状态图