

面向服务的软件架构设计——在线购物系统

一、实验目的及要求

通过实践一个高度分布的基于 Web 的在线购物系统的需求建模、需求分析、架构设计、详细设计的全过程，掌握面向服务的软件系统分析和设计方法。熟练使用 UML 语言及其相关建模方法，熟练使用 StartUML 建模工具进行相关设计开发。

二、实验设备（环境）及要求

1. PC 机最低配置：2G Hz 以上 CPU；1G 以上内存；1G 自由硬盘空间
2. StartUML

三、实验内容与步骤

1. 安装 StartUML；
2. 对“在线购物系统”开发用例模型，详细了解用例编档的方法；
3. 面向问题域的静态建模；
4. 识别系统中的各类对象；
5. 动态建模；
6. 面向服务的分层体系结构设计；
7. 服务通信及服务接口设计。

四、实验结果与数据处理

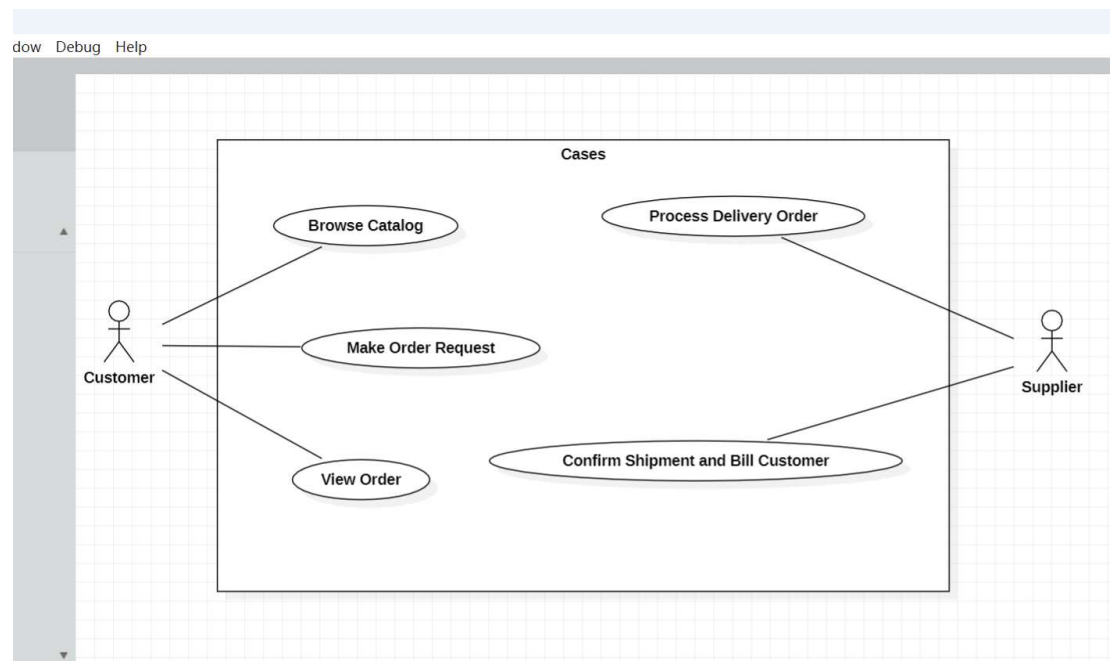
1 问题描述

在基于 Web 的“在线购物系统”中，客户可以向供应商请求购买一件或者多件商品。客户提供个人信息，例如地址和信用卡信息。这些信息被存储在客户账户中。如果信用卡是有效的那么系统创建一个配送订单并且发送给供应商。供应商

检查可用的库存，确认订单，并且输入一个计划好的配送日期。当订单完成配送后，系统通知客户并且向客户的信用卡账户收费。

2 用例建模

下图描绘了基于 Web 的“在线购物系统”的用例模型。有两个参与者：“客户 (Customer)”和“供应商” (Supplier)。客户浏览目录和请求购买商品，供应商提供目录和照务客户的购买请求。有三个用例是由客户发起的，它们是：“浏览目录” (Browse Catalog 客户浏览目录并挑选商品；“下单请求” (Make Order Request)，客户发出一个购买请求；“查看订单” (View Order)，客户查看订单详细信息。有两个用例是由供应商发起的，即“处理配送订单” (Process Delivery Order)，以满足客户的订单服务；“确认配送和给客户开账单 (Confirm Shipment and Bill Customer)，以完成购买过程。



2.1 “浏览目录”用例描述

用例描述：

见右文本框。

用例名称：浏览目录

概述：客户浏览万维网目录，从供应商的目录中查看各种各样的商品项，并且从目录中选择商品。

参与者：客户

前置条件：客户的浏览器链接到供应商的目录网站。

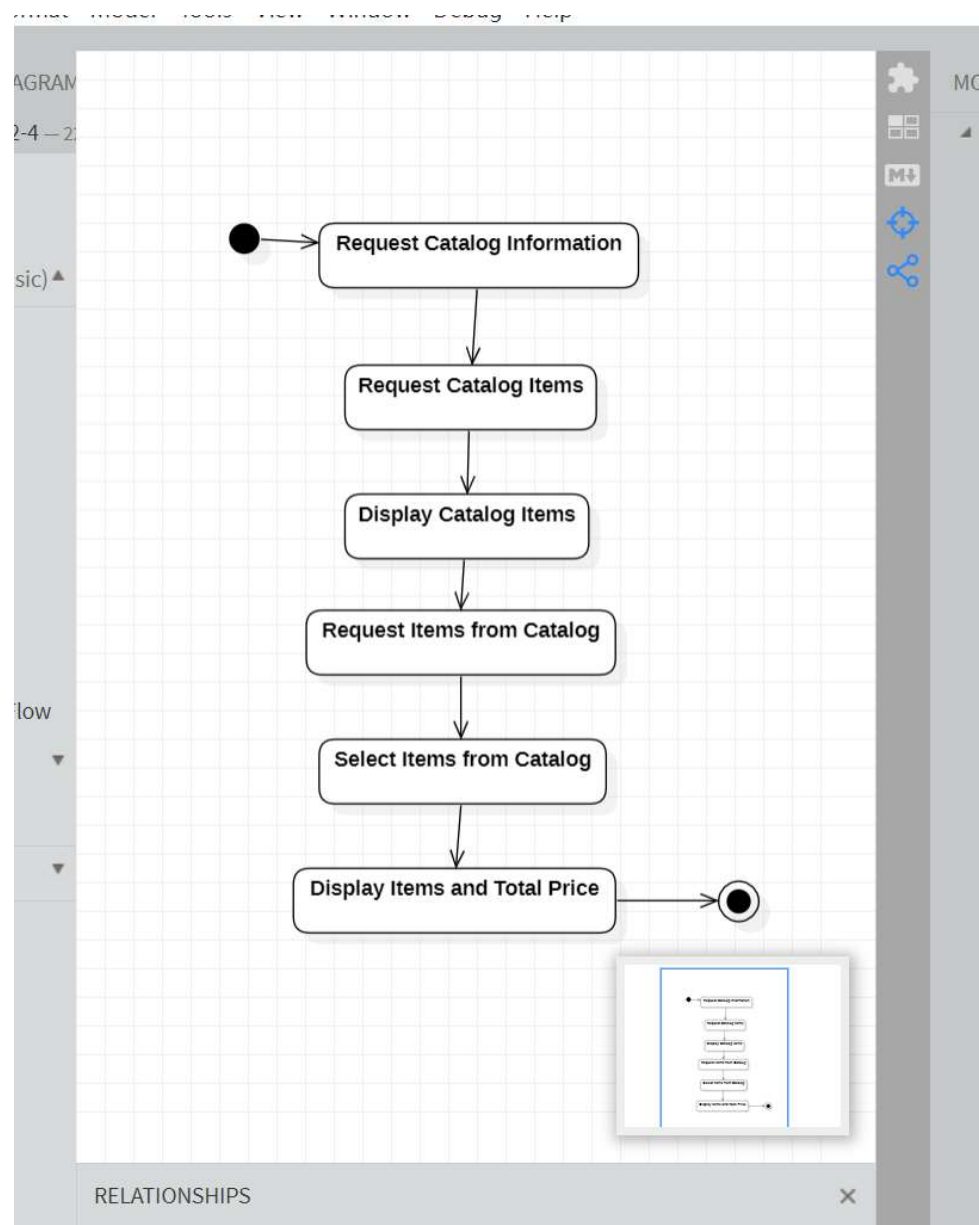
主序列：

1. 客户请求浏览目录。
2. 系统向客户显示目录信息。
3. 客户从目录中选择商品。
4. 系统显示商品列表，包含商品描述、价格以及总价。

可替换序列：步骤 3: 客户没有选择商品并且退出。

后置条件：系统显示了所选择的商品列表。

活动图：



2.2 “下单请求”用例描述

用例名称：下单请求

概述：客户输入——一个订单请求来购买目录商品。客户的信用卡被检查其有效性和是否有足够的额度来支付请求的目录商品。

参与者：客户

前置条件: 客户选择了一个或者多个目录商品。

主序列:

1. 客户提供订单请求和客户账户 ID 来支付购买。
2. 系统检索客户账户信息，包括客户的信用卡详细信息。
3. 系统根据购买总额检查客户信用卡，如果通过，创建一个信用卡购买授权号码。
4. 系统创建一个配送订单，包括订单细节、客户 ID 和信用卡授权号码。
5. 系统确认批准购买，并且向客户显示订单信息。

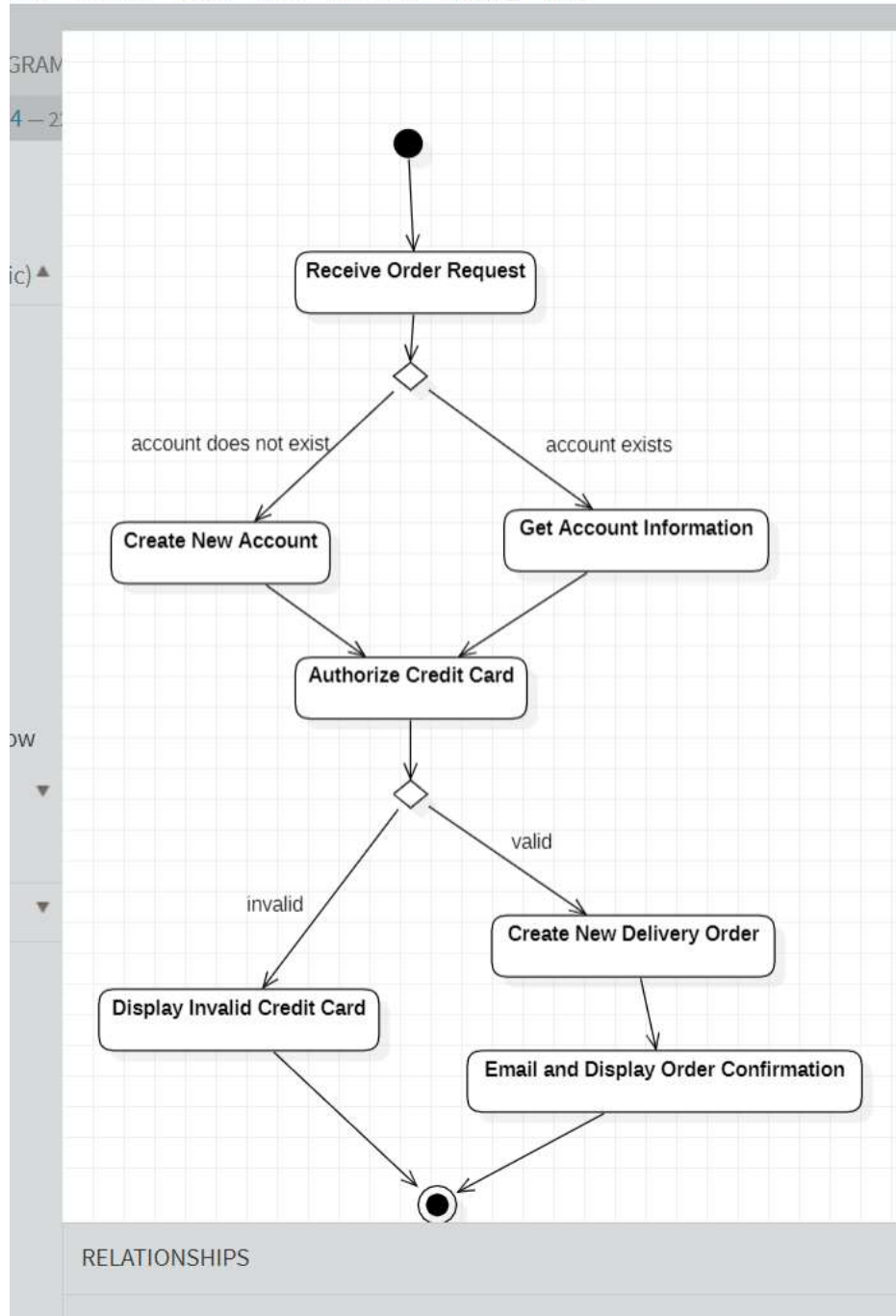
可替换序列:

步骤 2: 如果客户没有账户，系统提示客户提供信息来创建一个新账户。客户可输入账户信息或者取消订单。

步骤 3: 如果客户的信用卡授权被拒绝（例如，无效的信用卡或者客户的信用卡账户资金不足），系统提示客户输入一个不同的信用卡号码。客户可输入一个不同的信用卡号码或者取消订单。

后置条件: 系统为客户创建了一个配送订单。

用例活动图如下：



2.3 “处理配送订单”用例描述

用例名称: 处理配送订单

概述: 供应商请求一个配送订单; 系统确定库存对于满足订单是可用的, 并且显示订单。参与者: 供应商

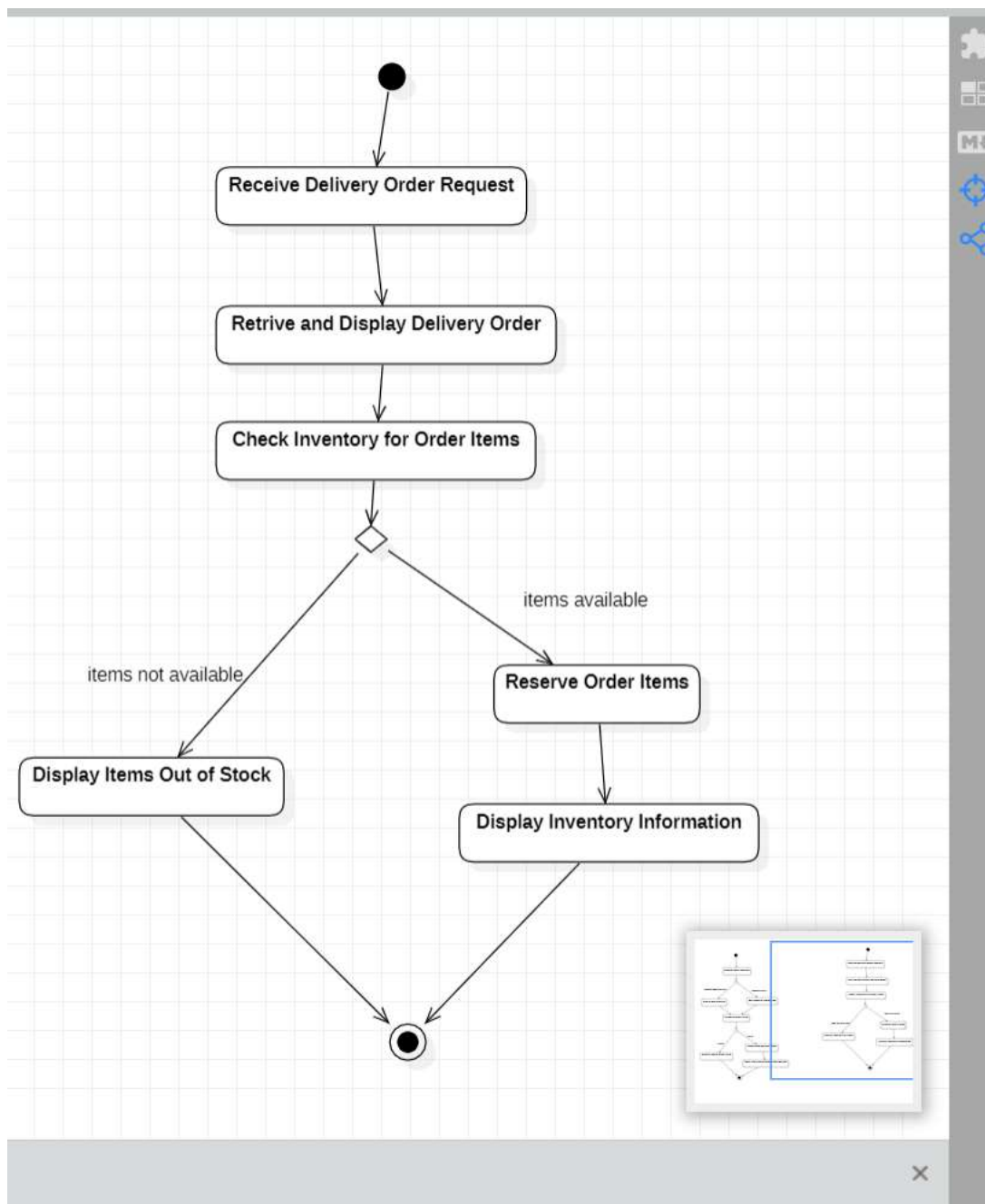
前置条件: 供应商需要处理一个配送订单并且一个配送订单存在。主序列:

1. 供应商请求下一个配送订单。
2. 系统检索并且显示配送订单。
3. 供应商为配送订单请求商品库存检查。
4. 系统确定库存中的商品对于满足订单是可用的，并且保留这些商品。
5. 系统给供应商显示库存信息，并且确认商品被保留。

可替换序列: 步骤 4: 如果商品库存不足，系统显示警告信息。

后置条件: 系统为配送订单保留了库存商品。

用例活动图如下:



2.4 “确认配送和给客户开账单”用例描述

用例名称: 确认配送和给客户开账单

概述: 供应商手工地准备配送并且确认配送订单已经准备好。系统通知客户订单正在配送。系统通过客户的信用卡收取购买商品的款项并且更新相关库存商品的库存。

参与者: 供应商

前置条件: 库存商品已经为客户的配送订单进行了预留。主序列:

1. 供应商手工地准备配送并且确认配送订单已经准备好配送。2. 系统检索客

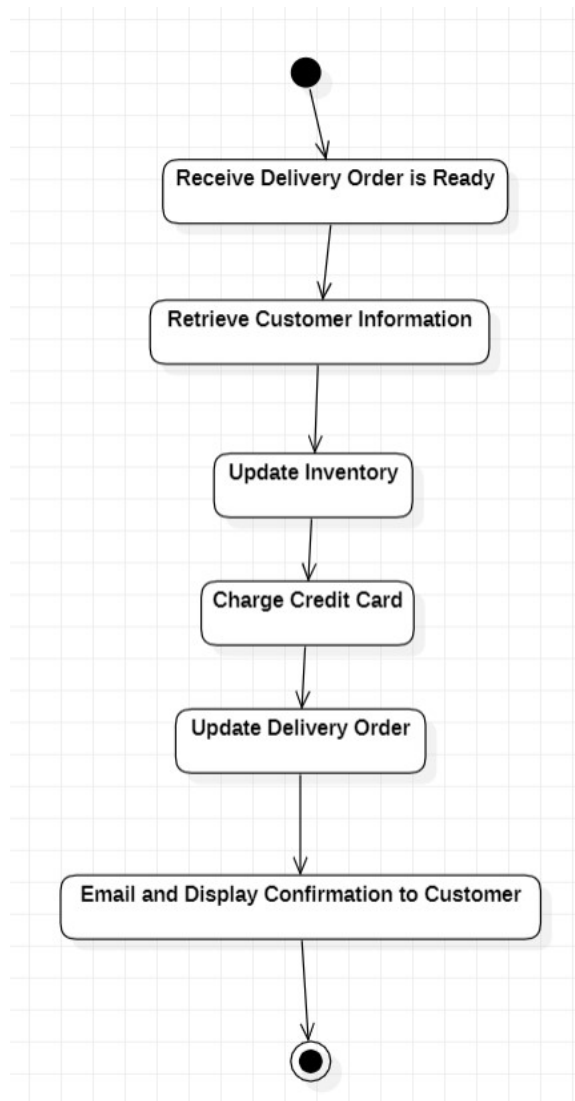
户的账户信息，包括发货单和客户的信用卡细节。3. 系统更新库存，确认购买。

4. 系统通过客户信用卡收取购买商品的款项并且创建一个信用卡收费确认号码。5. 系统用信用卡收费确认号码更新配送订单信息。

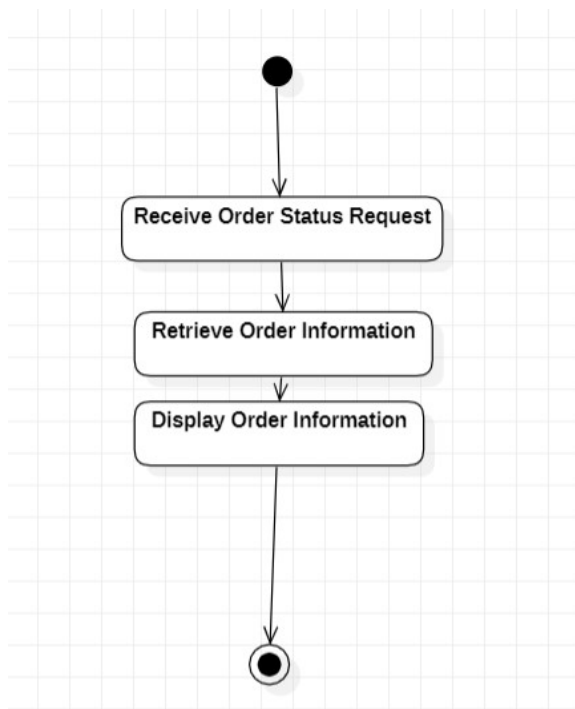
6. 系统给客户发送确认邮件。

7. 系统向供应商显示确认信息来完成配送订单的配送。

后置条件: 系统提交了库存，向客户收费，并且发送了确认信息。

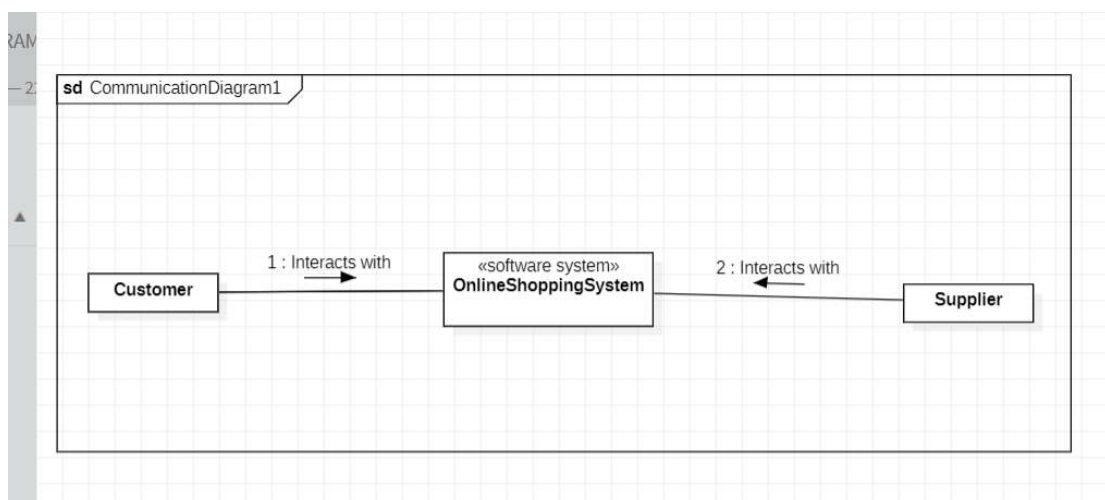


2.5 “查看订单”用例活动图



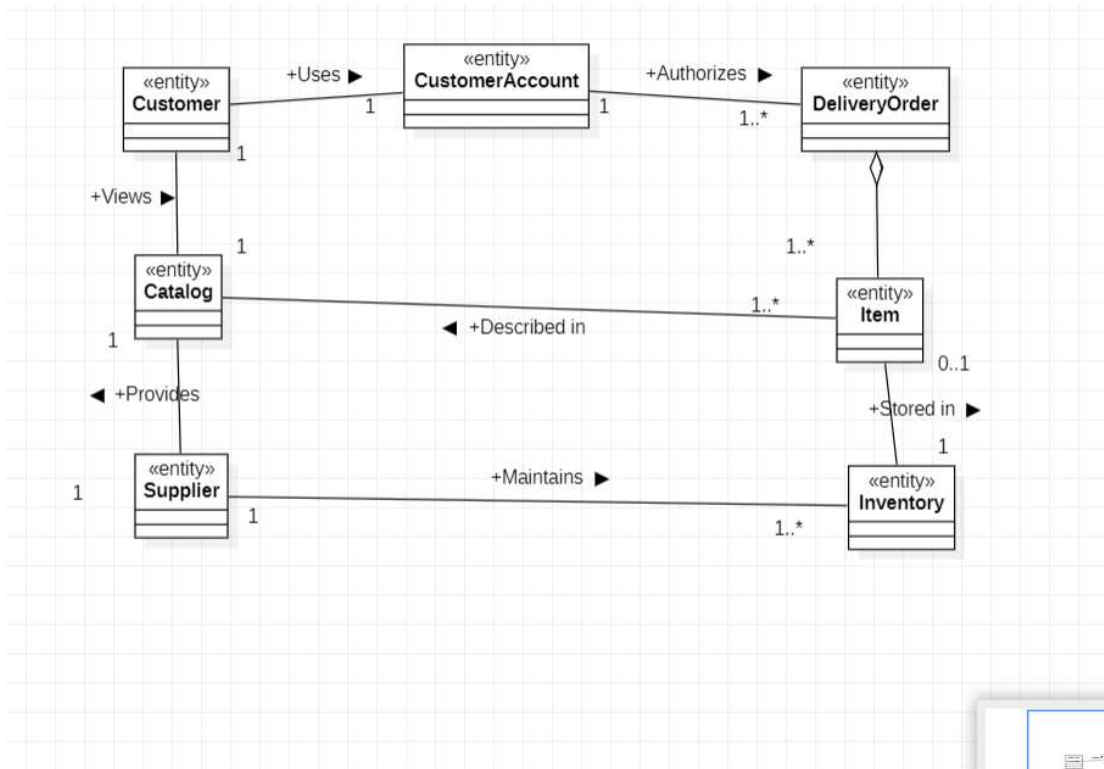
3 静态建模

3.1 软件系统上下文建模

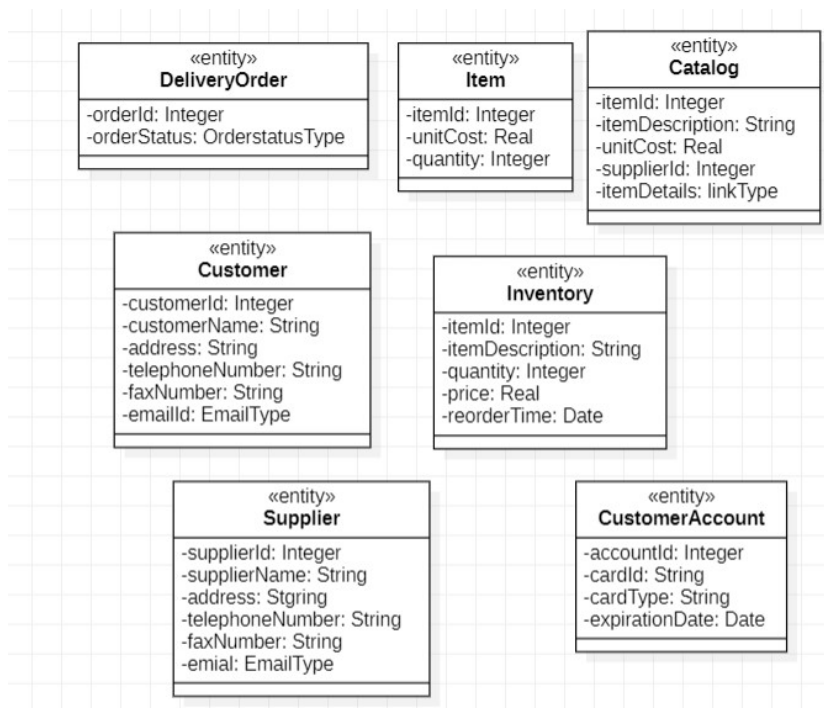


3.2 问题域的静态实体类建模

静态建模如图所示：

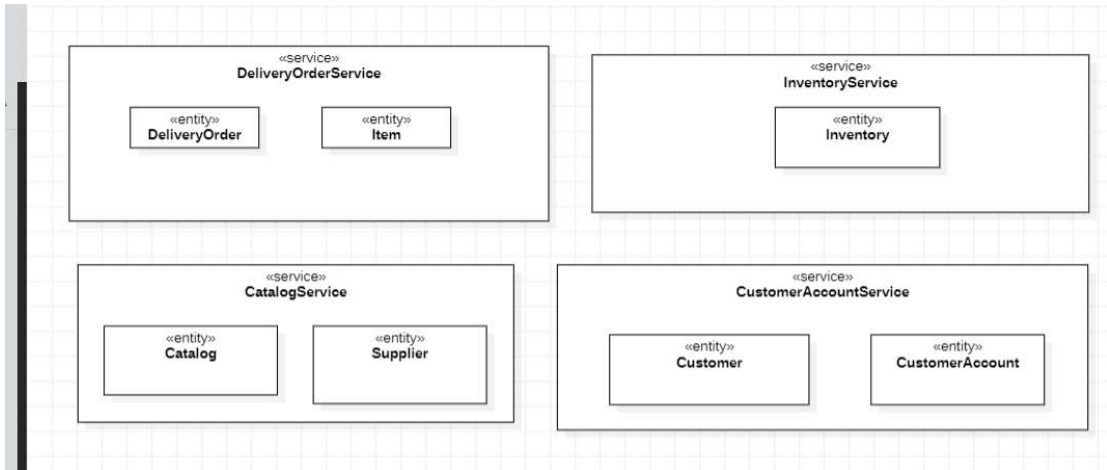


实体类如下图所示：

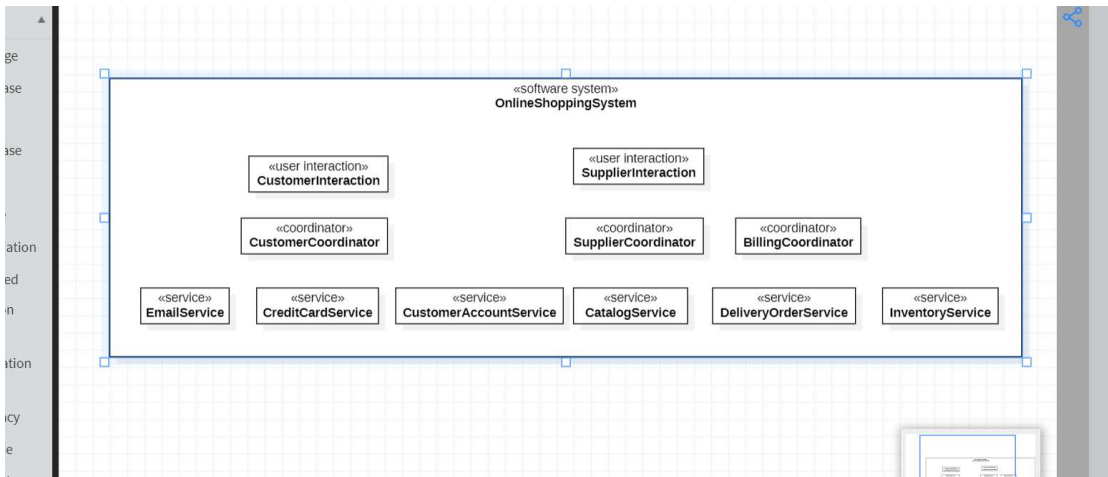


4. 对象和类组织

在线购物系统的服务类和实体类如下图所示：



在线购物系统的类组织如下图所示：



5. 动态建模

5.1 “浏览目录”用例的动态建模

消息描述如下所示：

B1: 客户通过“客户交互”发出一个目录请求。

B2: “客户协调者”被实例化来帮助客户。在客户请求的基础上，“客户协调者”为客户选择一个目录来浏览。

B3: “客户协调者”向“目录服务”请求信息。B4: “目录服务”发送目录信息给“客户协调者”。B5: “客户协调者”把信息转发给“客户交互”。B6: “客户交互”向客户显示目录信息。

B7: 客户通过“客户交互”选择一个目录。

B8: “客户交互”传递请求给“客户协调者”。

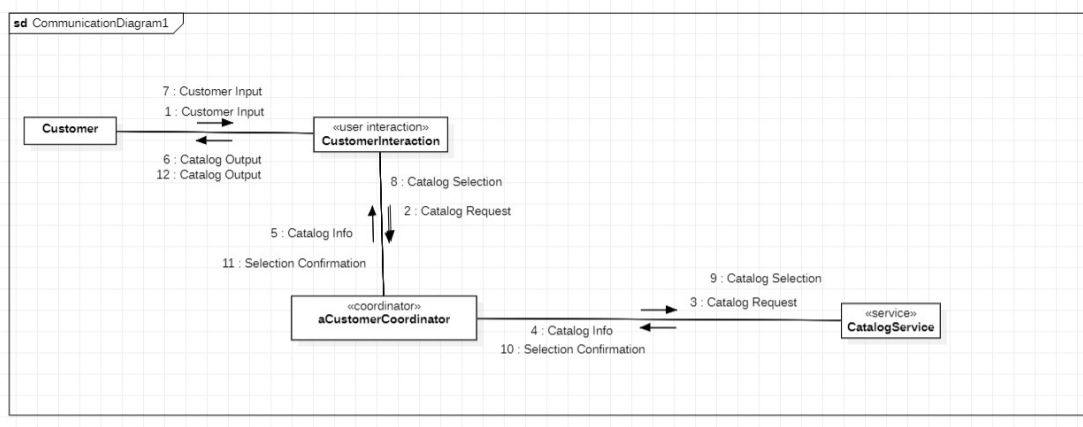
B9: “客户协调者”向“目录服务”请求目录选择。

B10: “目录服务”确认目录商品的可用性并且发送商品价格给“客户协调者”。

B11: “客户协调者”转送信息给“客户交互”。

B12: “客户交互”向客户显示目录信息，包括商品价格和总价。

通信图如下图所示：



5.2 “下单请求”用例的动态建模

M1: 客户向“客户交互”提出订单请求。

M2: “客户交互”将订单请求发送给“客户协调者”。

M3, M4 : “客户协调者”发送账户请求给“客户账户服务”，并且接收账户信息，包括客户的信息卡详细信息。

M5: “客户协调者”向“信用卡服务”发送客户的信用卡信息和付款授权请求（这相当于一个“准备提交”（Prepare to Commit）的消息）。

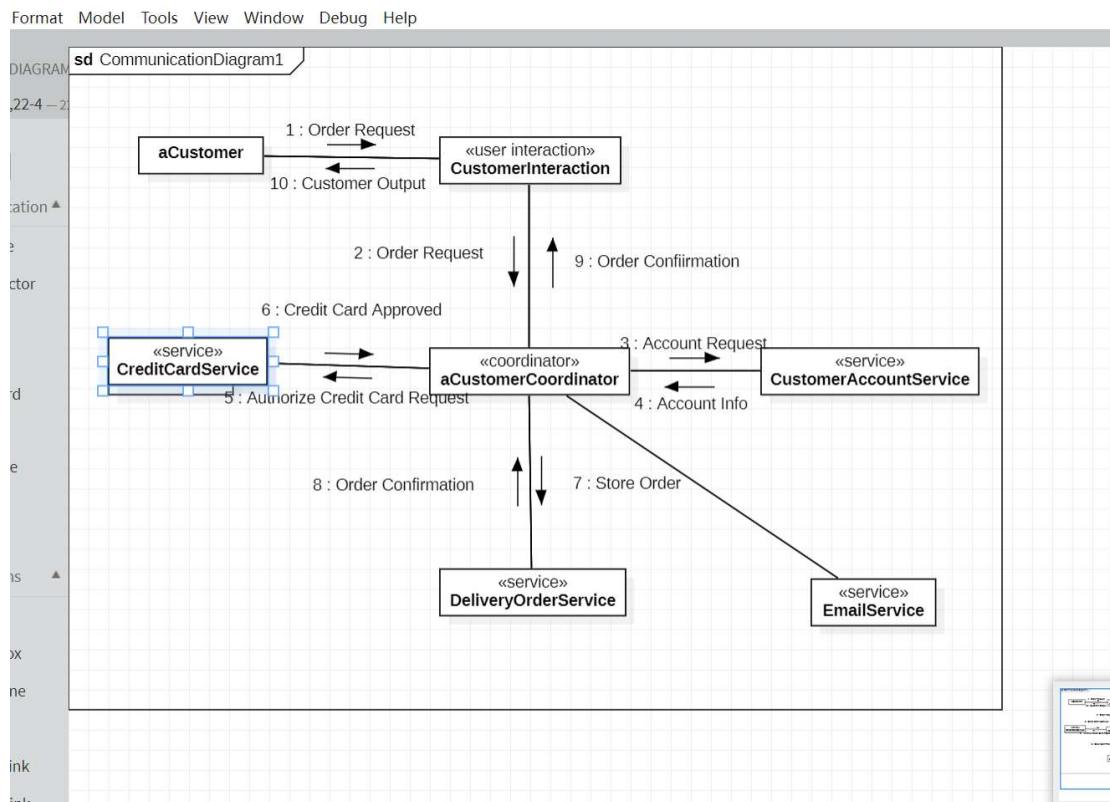
M6: “信用卡服务”向“客户协调者”发送一个信用卡批准（这相当于“准备好提交”（Ready to Commit）的消息）。

M7, M8: “客户协调者”发送订单请求给“配送订单服务”。

M9, M9a : “客户协调者”发送订单确认给“客户交互”，并且通过“电子邮件服务”向客户发送一封订单确认的邮件。

M10: “客户交互”向客户输出订单确认。

通信图如下图所示：



5.3 “处理配送订单”用例动态建模

D1: 供应商请求一个新的配送订单。

D2: “供应商交互”向“供应商协调者”发送供应商的请求。D3: “供应商协调者”请求“配送订单服务”选择一个配送订单。D4: “配送订单服务”发送配送订单给“供应商协调者”。

D5: “供应商协调者”请求检查商品库存。

D6: “库存服务”返回商品信息。

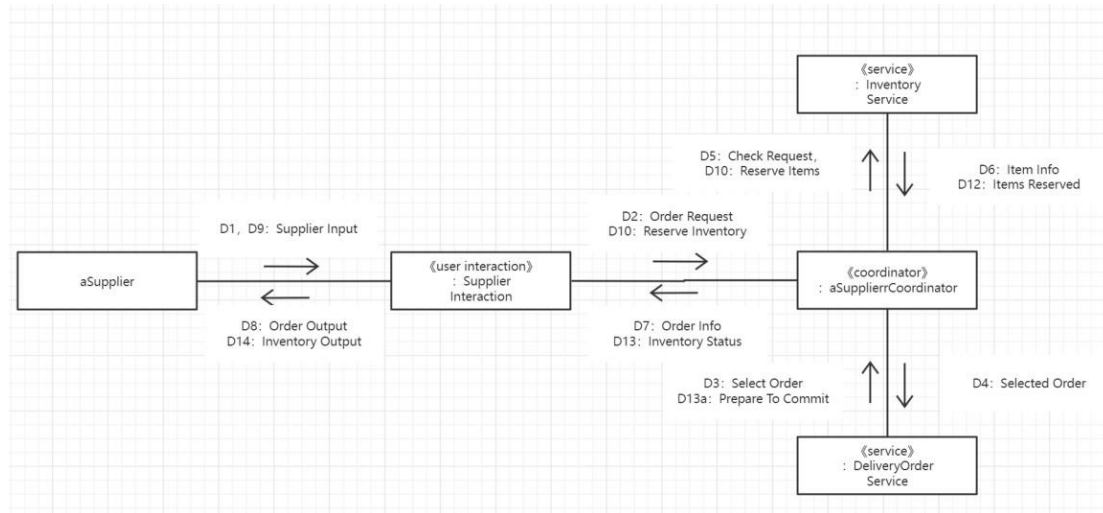
D7: “供应商协调者”发送订单信息给“供应商交互”。D8: “供应商交互”向供应商显示配送订单的信息。D9: 供应商请求系统在库存中保留商品。

D10: “供应商交互”发送供应商的请求给“供应商协调者”来保留库存。

D11: “供应商协调者”请求“库存服务”来保留库存中的商品（这相当于“准备提交”的消息）。

D12: “库存服务”向“供应商协调者”确认商品的保留（这相当于“准备好提交”的消息）。

D13: “供应商协调者”发送库存状态给“供应商交互”。D14: “供应商交互”向供应商显示库存信息。



5.4 “确认配送和给客户开账单”用例建模

S1: 供应商输入配送信息。

S2: “供应商交互”发送“准备好配送”的请求给“供应商协调者”。

S3: “供应商协调者”发送“订单准备好配送”（Order Ready for Shipment）的消息给“账单协调者”。

S4: “账单协调者”发送“准备提交”订单给“配送订单服务”。

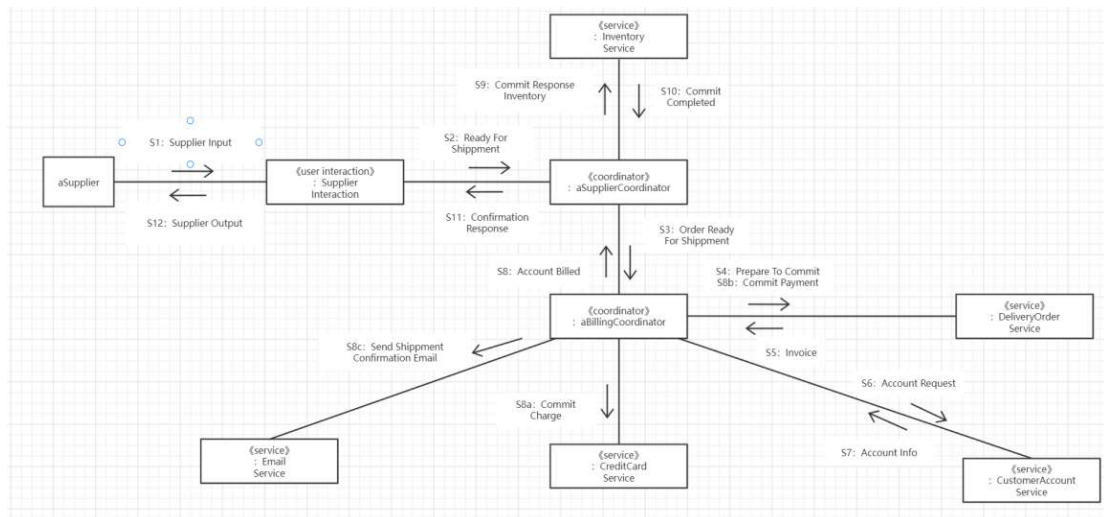
S5: “配送订单服务”回复“准备好提交”的消息和发货单，包括订单号、账户号和总价。

S6, S7: “账单协调者”发送账户请求给“客户账户服务”，“客户账户服务”返回账户信息。

S8, S8a, S8b, S8c : “账单协调者”发送“提交收费”(Commit Charge)的消息给“信用卡服务”,发送“提交支付”(commit Payment)的消息给“配送订单服务”,通过“电子邮件服务”发送确认邮件给客户,发送“账户已开账单”(Account Billed)的消息给“客户协调者”。

s9, S10: “供应商协调者”发送“提交库存”的消息给“库存服务”,“库存服务”返回提交已完成。

S11, S12: “供应商协调者”发送确认响应给“供应商交互”,“供应商交互”接着发送配送确认消息给供应商。



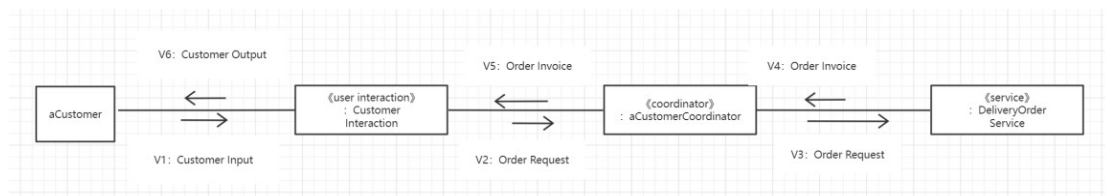
5.5 “查看订单”用例建模

V1, V2:客户通过“客户交互”发出一个订单发货单的请求。

V3: “客户协调者”向“配送订单服务”发出一个订单请求。

V4: “配送订单服务”发送订单发货单信息给“客户协调者”。V5: “客户协调者”转送信息给“客户交互”。

V6: “客户交互”向客户显示订单信息。



6. 面向服务体系结构的代理者和包装器技术支持

“在线购物系统”中使用了多个遗留数据库。静态模型中的许多实体类代表了存储在遗留数据库中的持久化数据，每个遗留数据库都是一个驻留在大型机上的单机数据库，这些数据库需要通过代理者和包装器技术整合到应用中。

尽管存在着不同的遗留数据库，但对象代理者和包装器技术提供了一种系统的方法来把异构的遗留数据库整合为一个面向服务的体系结构。在供应商组织内的遗留数据库包括目录数据库、库存数据库、客户账户数据库和配送订单数据库。

数据库包装器类被设计用来为遗留数据库提供一个面向对象的接口，这个接口隐藏了如何读取和更新个体数据库的细节。为了把这些数据库整合到在线购物应用中，服务类被设计为通过数据库包装器类来访问遗留数据库。

7. 设计建模

7.1 面向服务的体系结构概述

在面向服务的体系结构中，服务通过代理者注册它们的服务名字和位置。这样，客户端就可以使用“服务发现”模式（也被认为是黄页）来发现新的服务，从而向代理者查询给定类型的服务。客户端可以选择一个服务，并且发送一个白页请求给代理者。

基于分层抽象体系结构模式，“在线购物系统”可以被设计为一个分层的体系结构。软件体系结构由三层组成：一个服务层、一个协调者层和一个用户交互层。此外，因为这个系统需要是高度灵活和分布式的，所以决定设计为一个面向服务的体系结构。在这种体系结构中，分布式构件能够发现服务并且与它们通信。

每个构件通过构件构造型（它是什么类型的构件，按照构件组织准则中所定义的构件类型）来描述。构件和服务接口设计是通过分析每个用例的通信图来确定的。

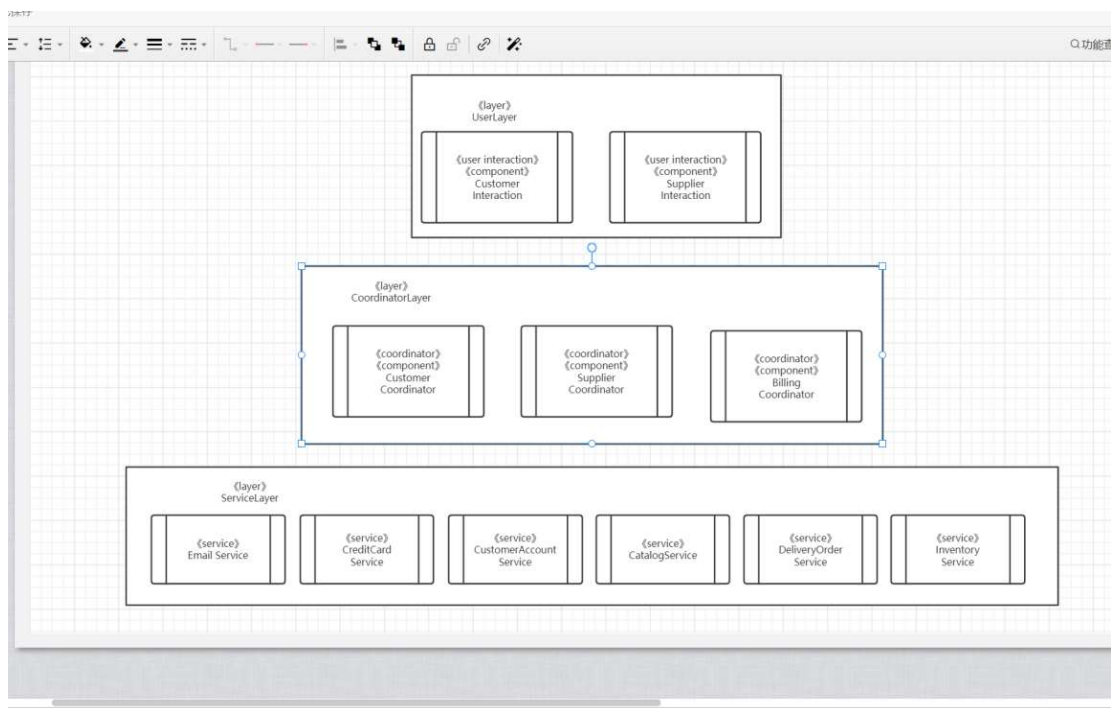
7.2 分层软件体系结构

层次 1:服务层(Service Layer)。共有 6 个服务，4 个是应用的一部分，2 个是外部服务。应用服务包括“目录服务”(Catalog Service)、“配送订单服务”(Delivery Order Service)，“库存服务”(Inventory Service)和“客户账户服务”(Customer Account Service)。外部服务包括用于向客户收取购买费用的“信用卡服务”(Credit Card Service)以及给客户发邮件消息的“电子邮件服务”(Email Service)。其中，每个信用卡公司都有一个服务，如万事达(Mastercard)和维萨(Visa)。

层次 2:协调层(Coordination Layer)。共有 3 个协调者构件：“供应商协调者”(SupplierCoordinator)、“客户协调者”(Customer Coordinator)和“账单协调者”(Billing Coordinator)。

层次 3：用户层(User Layer)。共有 2 个用户交互构件：“供应商交互”(SupplierInteraction)和“客户交互”(Customer Interaction)。

分层体系结构如下图所示



7.3 体系结构通信模式

为了处理软件体系结构中构件之间的多种通信方式，这里应用了以下几种通信模式：

带回复的同步消息通信。这是典型的面向服务的体系结构的通信模式，当客户端需要服务的信息并且在接受响应之前不能继续执行的时候使用这个模式。该模式被用于用户交互客户端和协调者之间，也被用于协调者和各种服务之间。

代理者句柄。每个服务向代理者注册服务信息，包括服务名称、服务描述和位置。“代理者句柄”模式允许客户查询代理者来确定它们应该连接的服务。

服务发现。“服务发现”模式被服务请求者使用来发现新的服务。它们能够用于发现新的可浏览目录。

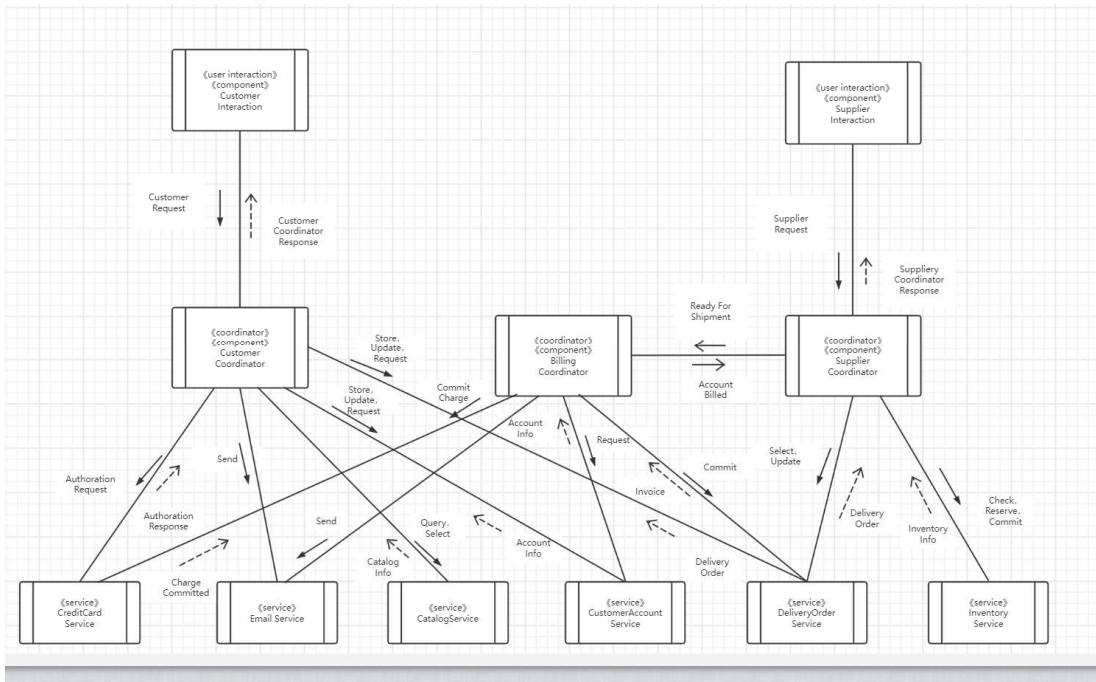
双向异步消息通信。这个模式被用于“供应商协调者”和“账单协调者”之间的双向异步通信。

两阶段提交。这个模式被用于确保对库存、信用卡和配送订单的更新是原子

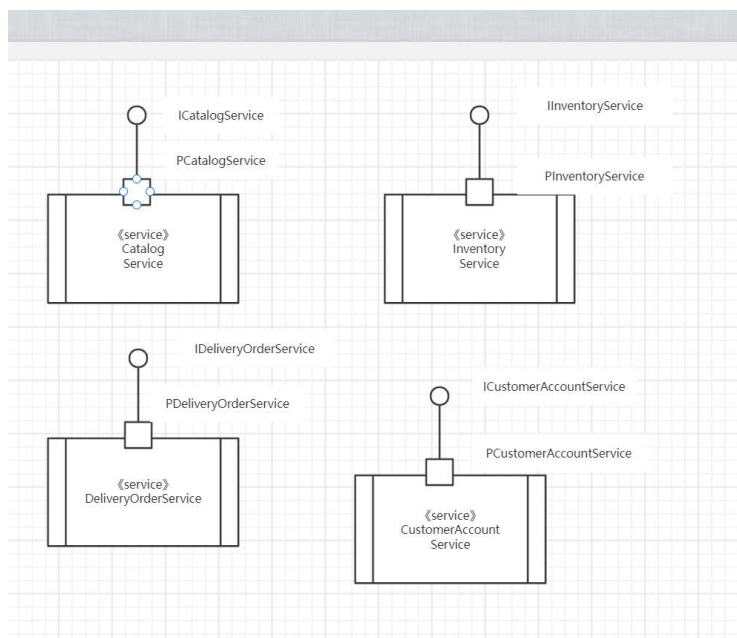
的，即所有的更新不是被提交就是被中止。

7.4 并发软件设计

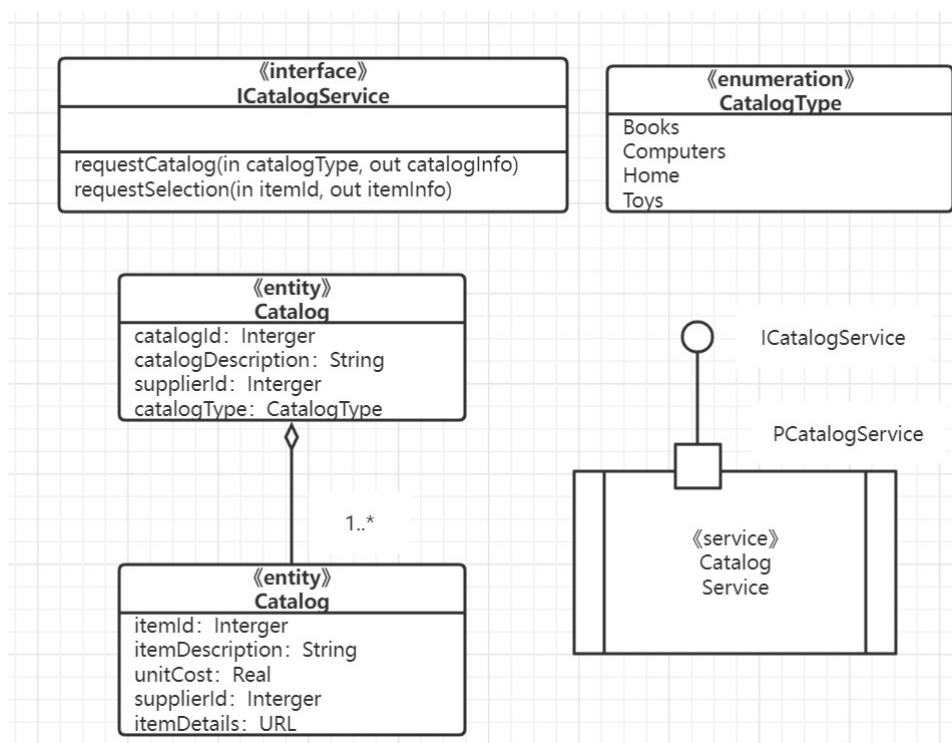
并发通信图如下图所示：



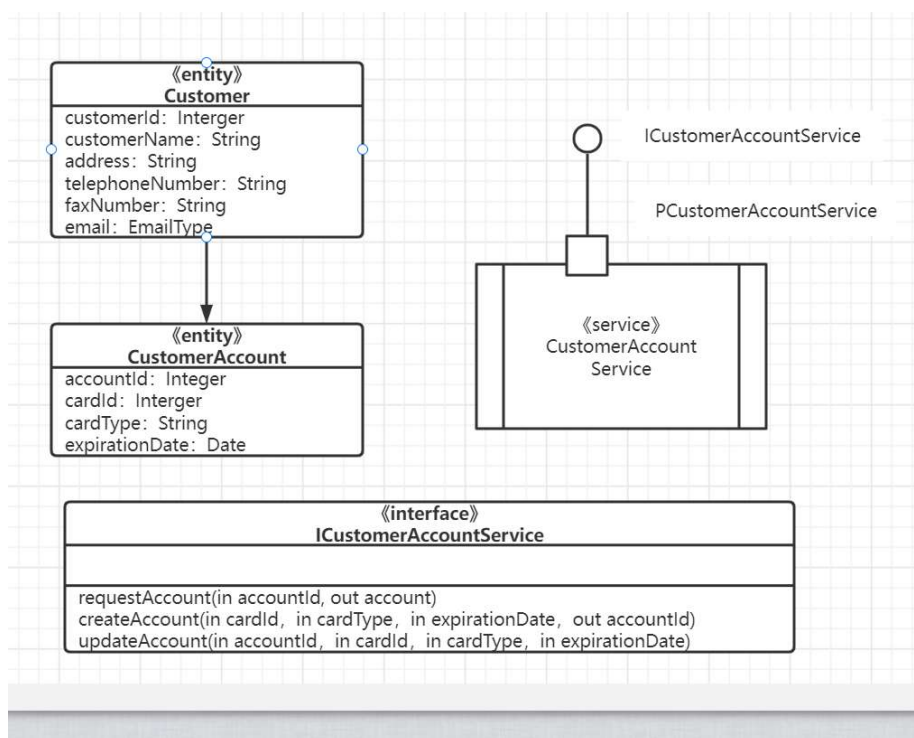
7.5 服务接口设计



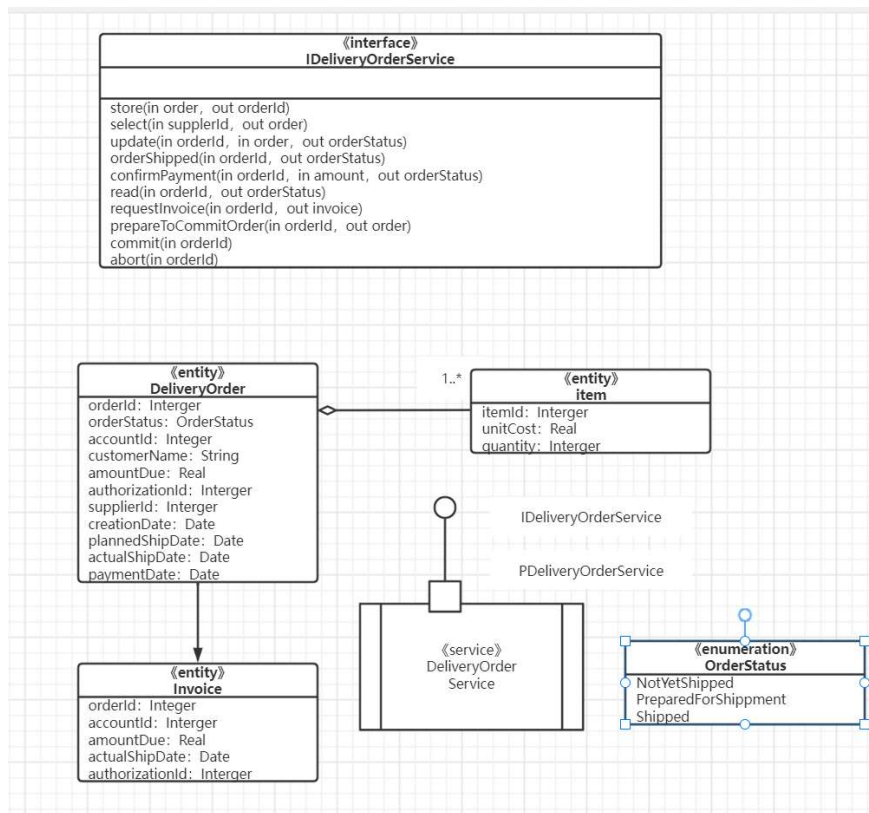
“目录服务”的服务接口如下图所示：



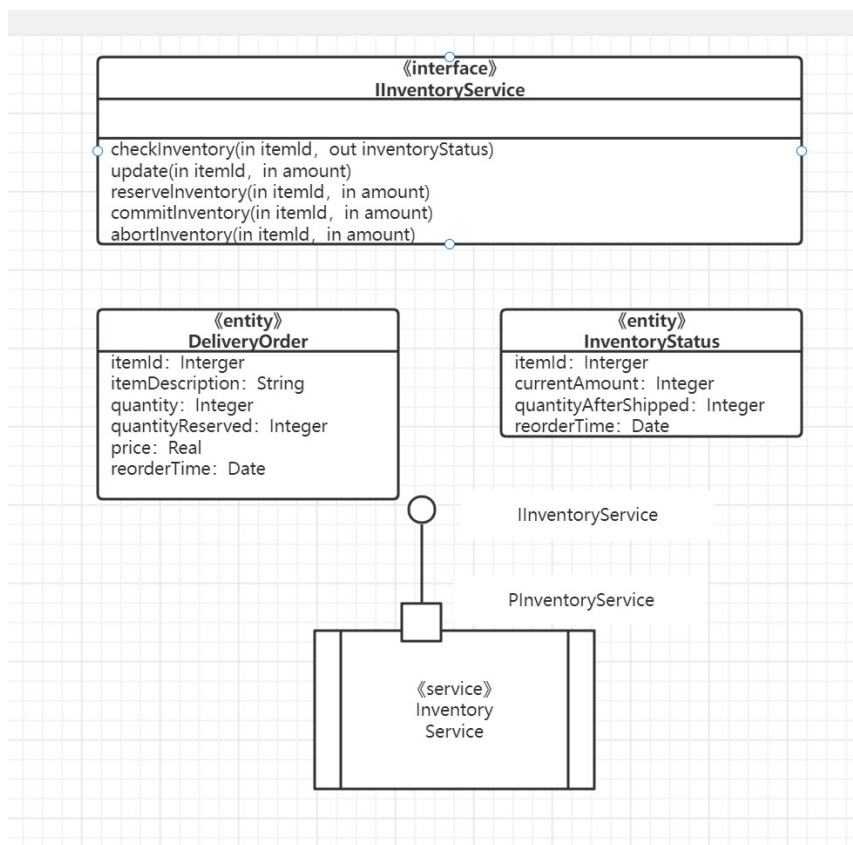
“客户账户服务”的服务接口如下图所示：



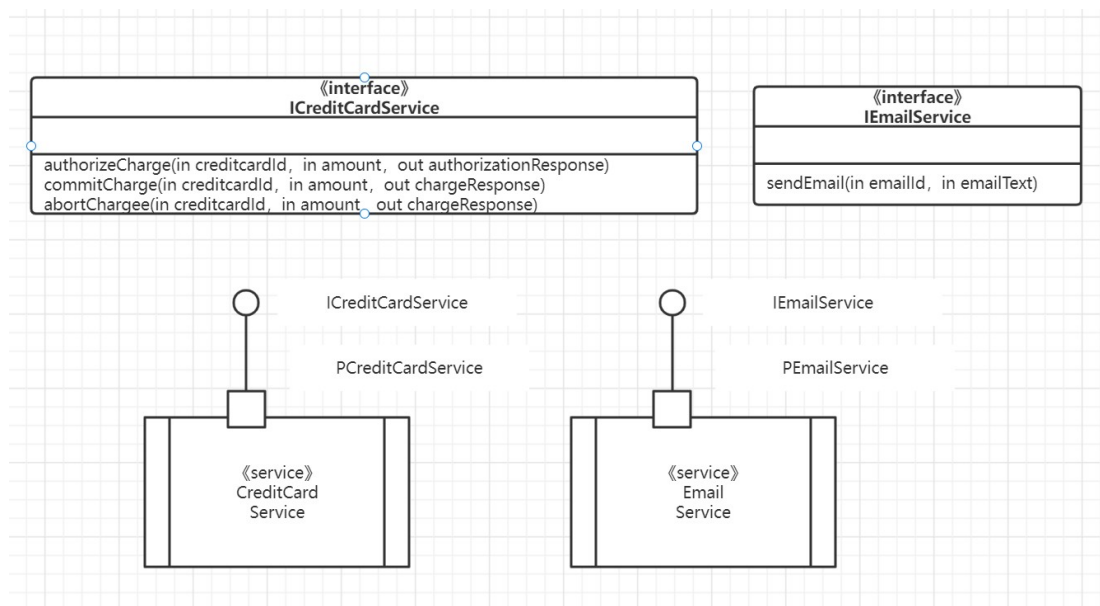
“配送订单服务”的服务接口如下图所示：



“库存服务”的服务接口如下图所示：

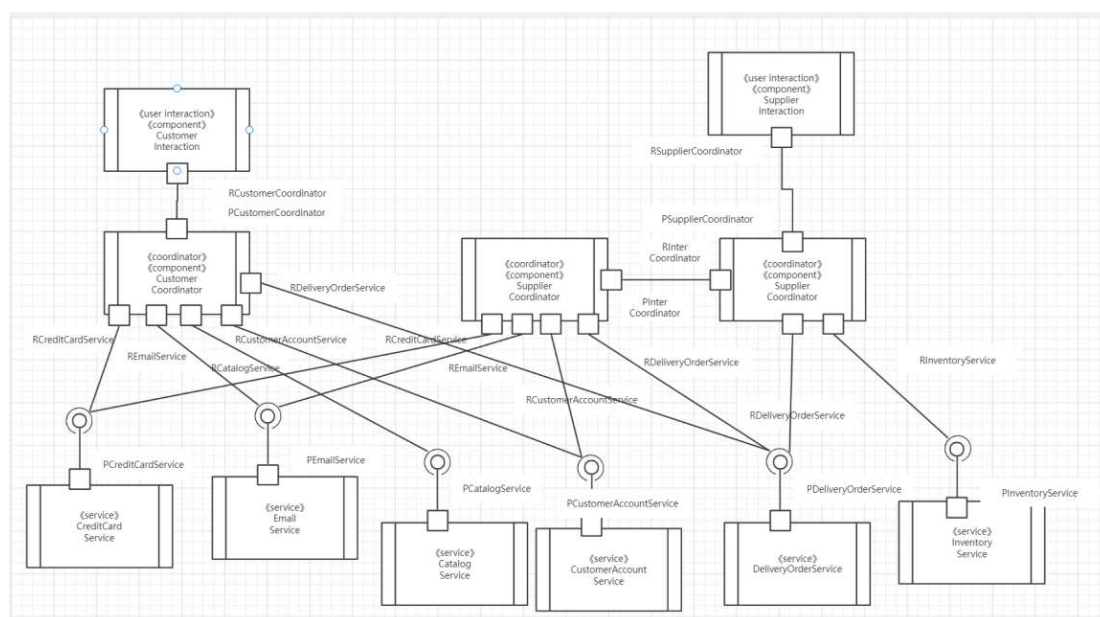


“信用卡服务”和“电子邮件服务”的服务接口如下图所示：



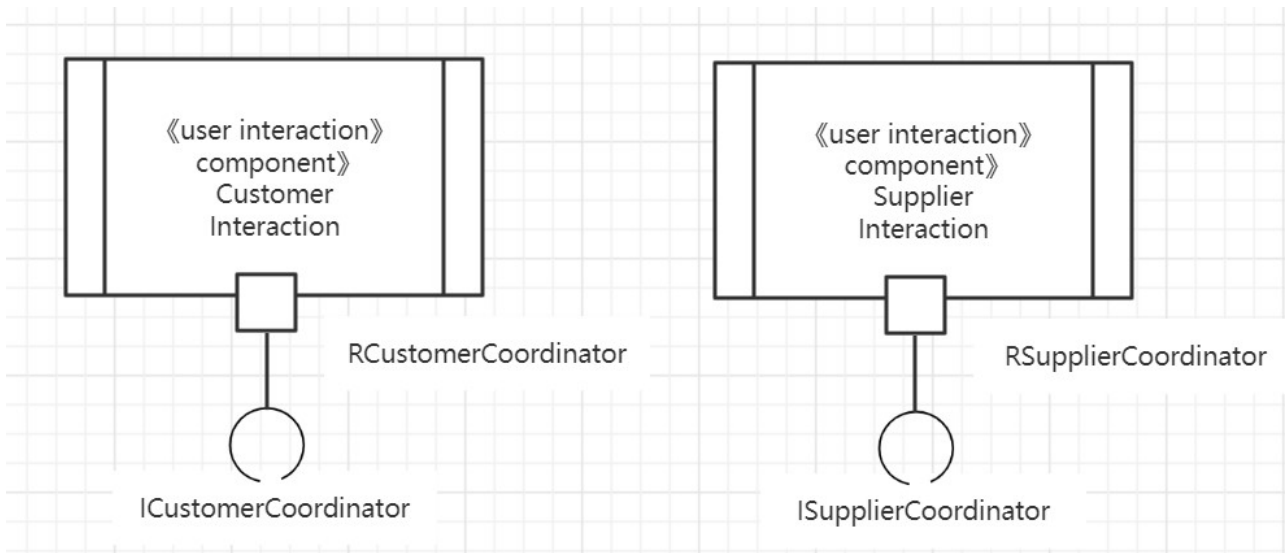
7.6 面向服务的软件体系结构设计

“在线购物系统”的面向服务的体系结构如下图所示：

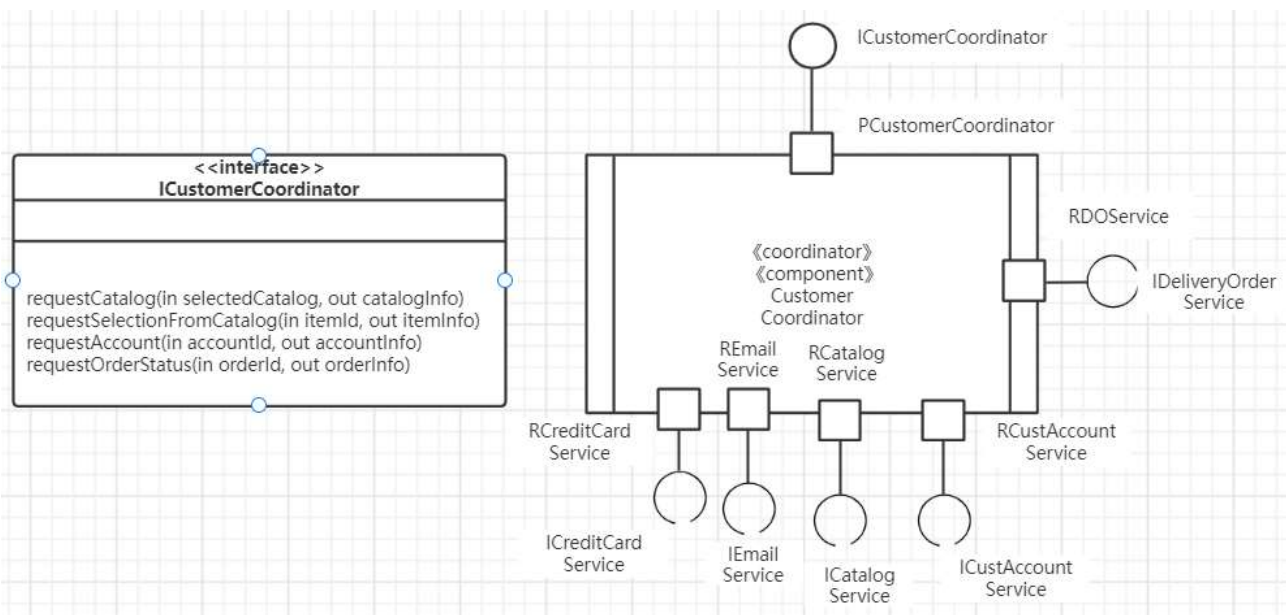


7.7 构件端口和接口设计

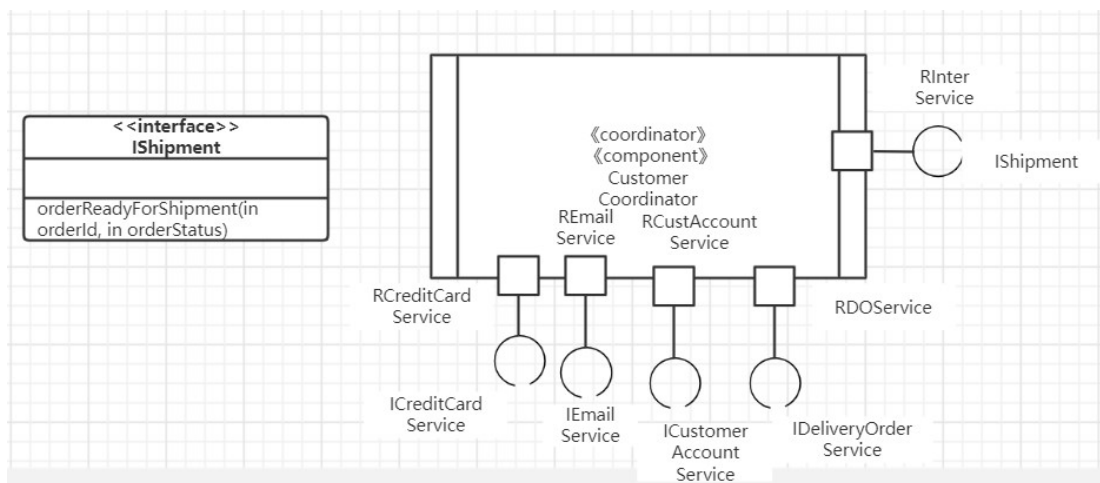
“客户交互”和“供应商交互”的构建端口和接口如下图所示：



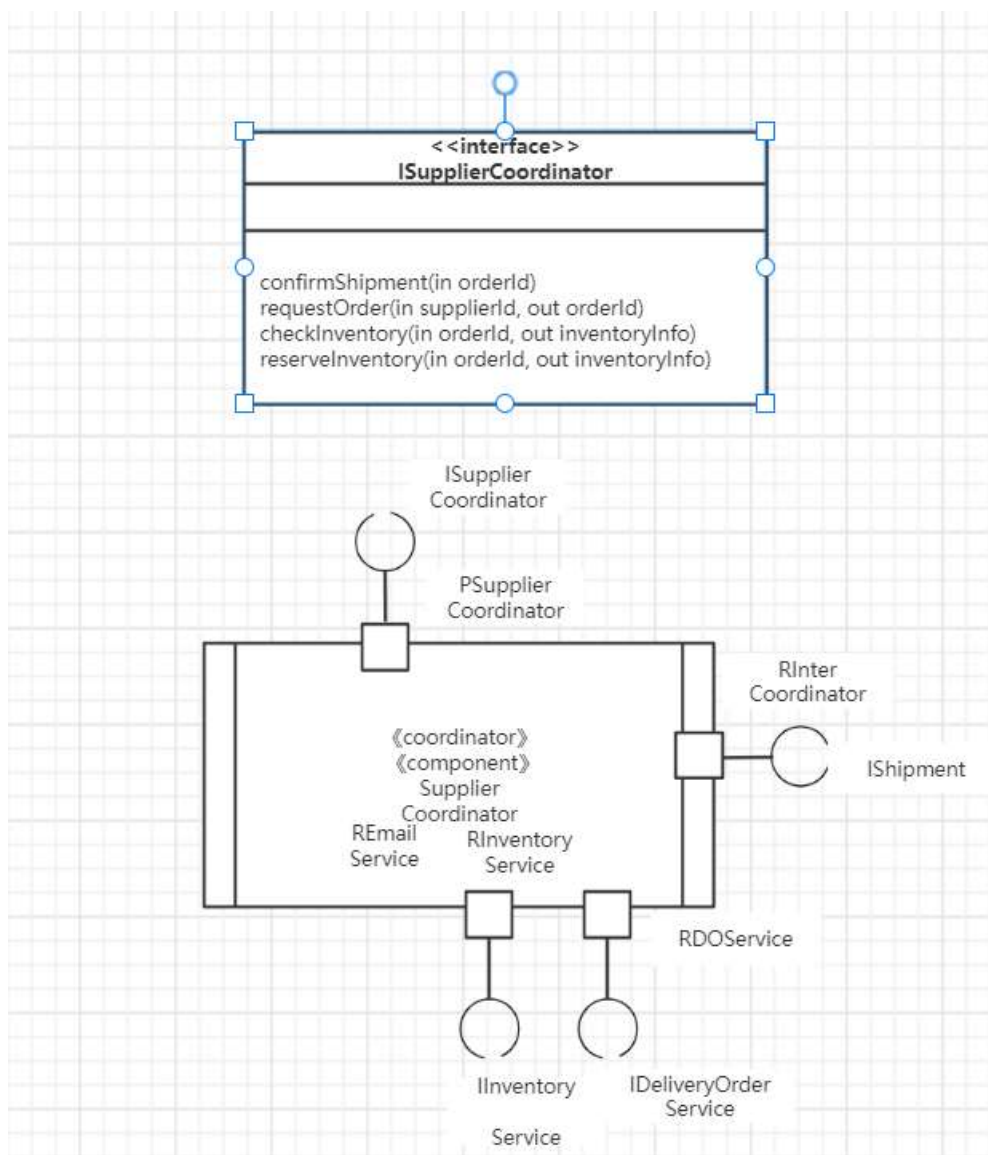
“客户协调者”构建端口和接口如下图所示：



“供应商协调者” 构件端口和接口如下图所示：



“账单协调者” 构建端口和接口如下图所示：



五、分析与讨论

1. 协调者类的主要作用是什么，它与服务类的区别有哪些？

协调者类是做出总体决策的类，它确定了相关对象集合的总体顺序安排。经常需要协调者类为用例的执行提供总体的顺序安排。它做出总体决策，并决定其他对象何时、以何种顺序参与用例中。协调者类根据它接收到的输入做出决策，并且不是状态相关的。因此，由协调者类发起的动作只取决于包含在传入消息中的信息，而不依赖于系统中之前发生的事情。

服务类是为其他对象提供服务的类，它们通常在面向服务的架构和应用中提供。客户端对象能从服务类请求一个服务，服务类将做出响应。服务类绝不会发起一个请求；然而，在响应服务请求时，它可能会寻求其他服务对象的帮助。服务类在面向服务的架构中扮演重要的角色，尽管它们也被用在其他的架构中，例如客户端/服务器体系结构和基于构件的软件体系结构。服务类可能封装了它需要用来服务客户端请求的数据，或者访问其他封装了该数据的实体对象。

2. 举例说明本次实验使用了哪些通信模式。

带回复的同步消息通信。这是典型的面向服务的体系结构的通信模式，当客户端需要服务的信息并且在接受响应之前不能继续执行的时候使用这个模式。该模式被用于用户交互客户端和协调者之间，也被用于协调者和各种服务之间。

代理者句柄。每个服务向代理者注册服务信息，包括服务名称、服务描述和位置。“代理者句柄”模式允许客户查询代理者来确定它们应该连接的服务。

服务发现。“服务发现”模式被服务请求者使用来发现新的服务。它们能够用于发现新的可浏览目录。

双向异步消息通信。这个模式被用于“供应商协调者”和“账单协调者”之间的双向异步通信。

两阶段提交。这个模式被用于确保对库存、信用卡和配送订单的更新是原子的，即所有的更新不是被提交就是被中止。

3. 本次实验中的供应商协调者类(SupplierCoordinator)使用了哪些请求接口，通过这些接口调用了哪些服务？

“供应商协调者”通过端口 `PSupplierCoordinator` 接收来自于“供应商交互”的供应商请求，这个端口支持一个供给接口 `ISupplierCoordinator`。“供应商协调者”是“库存服务”的一个客户端，并通过请求接口 `InventoryService` 与之通信。它像“客户协调者”一样是“配送订单服务”的客户端，并且有相同的请求接口 `IDeliveryOrderService`。“供应商协调者”也与“账单协调者”通信，当一个订单准备好配送和开账单时发送给它一个消息（S3）。这个通信的接口是 `IShipment`。

<p>六、教师评语</p> <p>签名:</p> <p>日期:</p>	<p>成绩</p>
-------------------------------------	-----------