Designing SOA
Case Study(*)

(*) Hassan Gomaa, *Software Modeling and Design: UML, Use Cases, Patterns, and Software Architectures*, Cambridge, 2011.

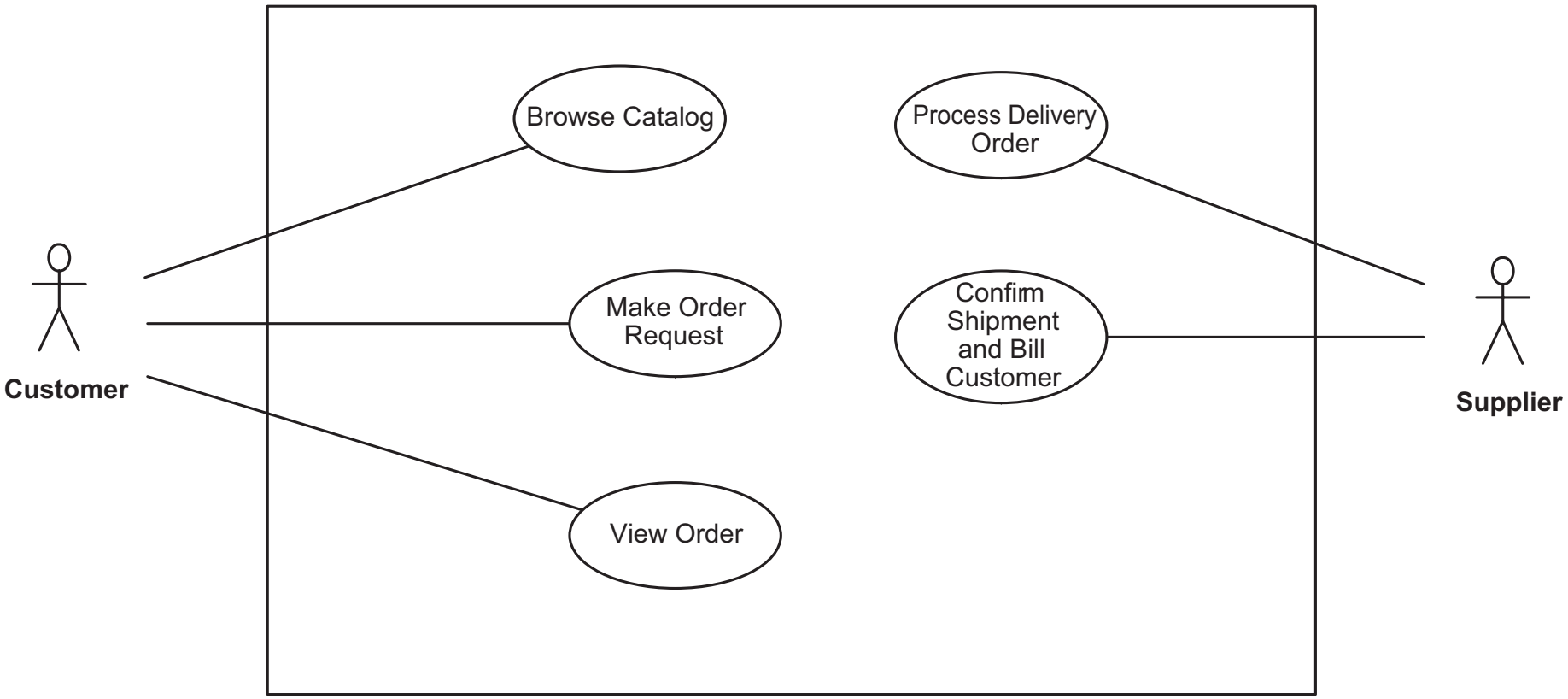
© UniRoma2 - Ingegneria del Software

Web-based Online Shopping System

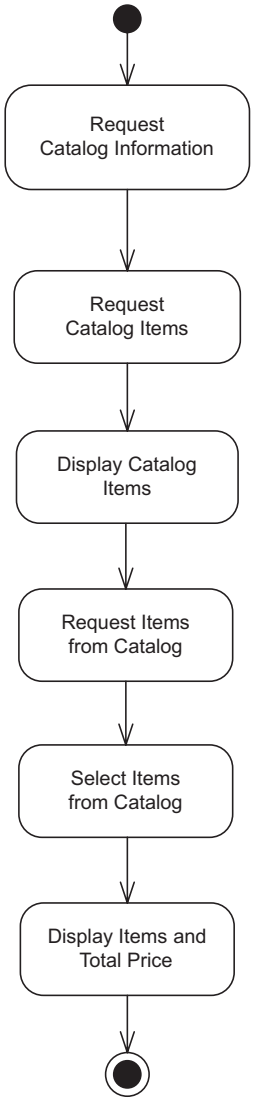
- In the **Web-based Online Shopping System**, customers can request to purchase one or more items from the supplier
- The customer provides personal details, such as address and credit card information
- This information is stored in a customer account
- If the credit card is valid, then a delivery order is created and sent to the supplier
- The supplier checks the available inventory, confirms the order, and enters a planned shipping date
- When the order is shipped, the customer is notified, and the customer's credit card account is charged

Use Case Modeling

Use Case Diagram

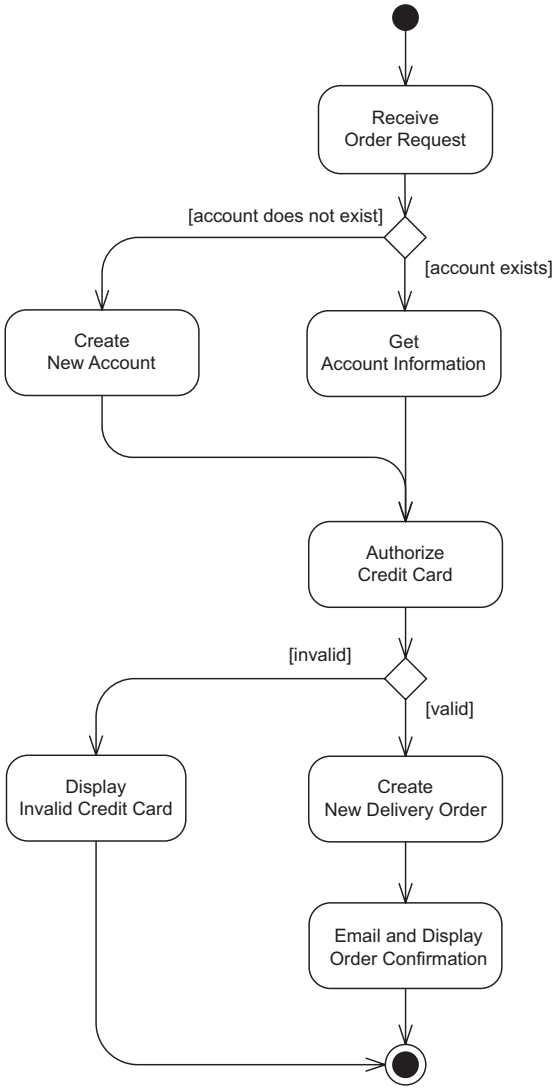


Activity diagram for *Browse Catalog* UC



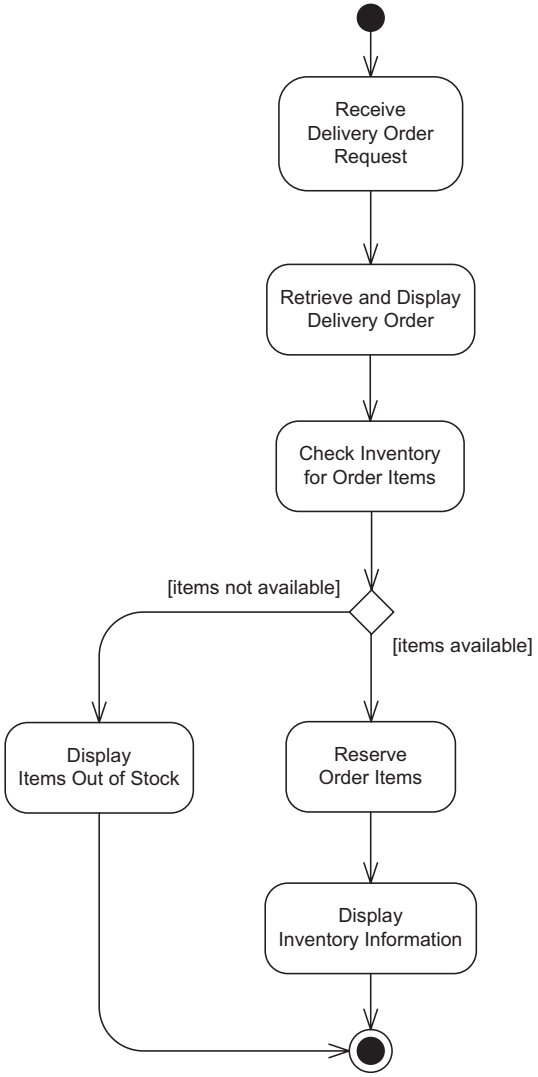
© UniRoma2 - Ingegneria del Software

Activity diagram for *Make Order Request* UC



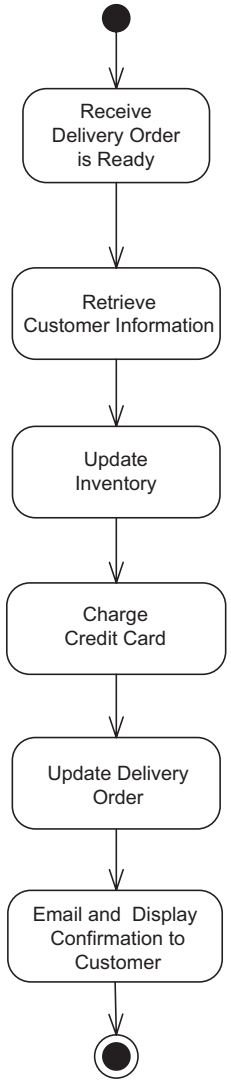
© UniRoma2 - Ingegneria del Software

Activity diagram for *Process Delivery Order* UC



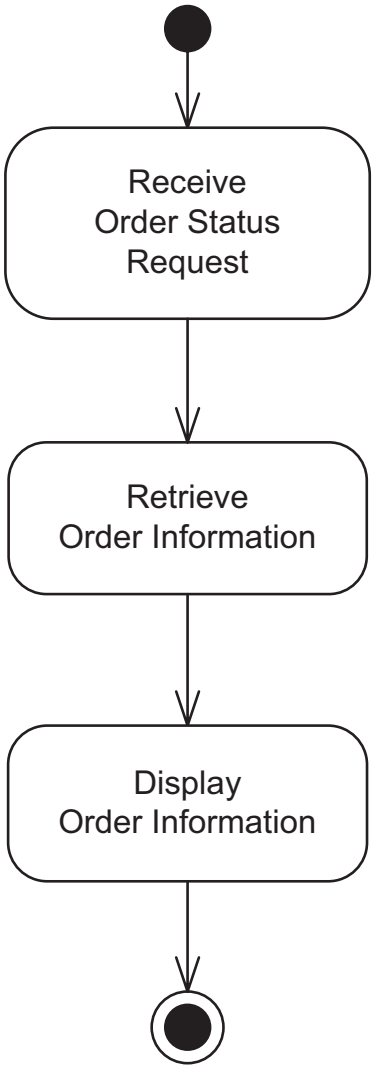
© UniRoma2 - Ingegneria del Software

Activity diagram for *Confirm Shipment and Bill Customer* UC



© UniRoma2 - Ingegneria del Software

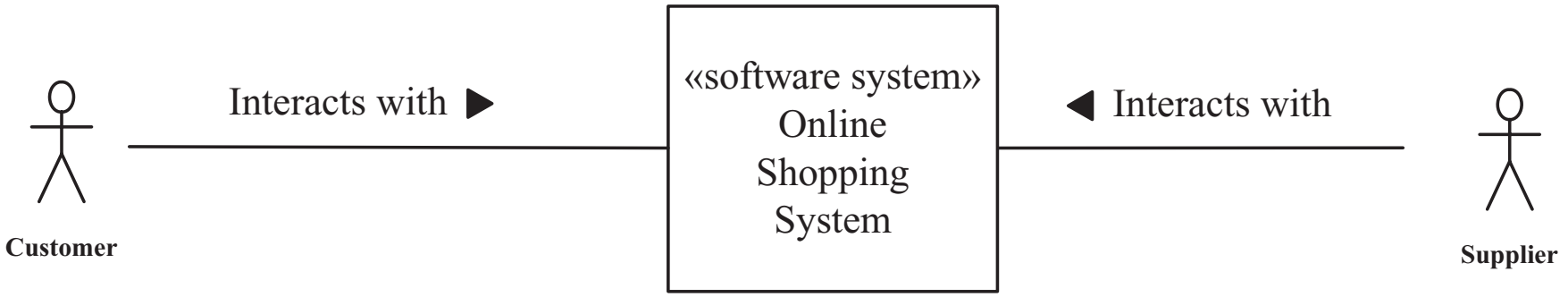
Activity diagram for View Order UC



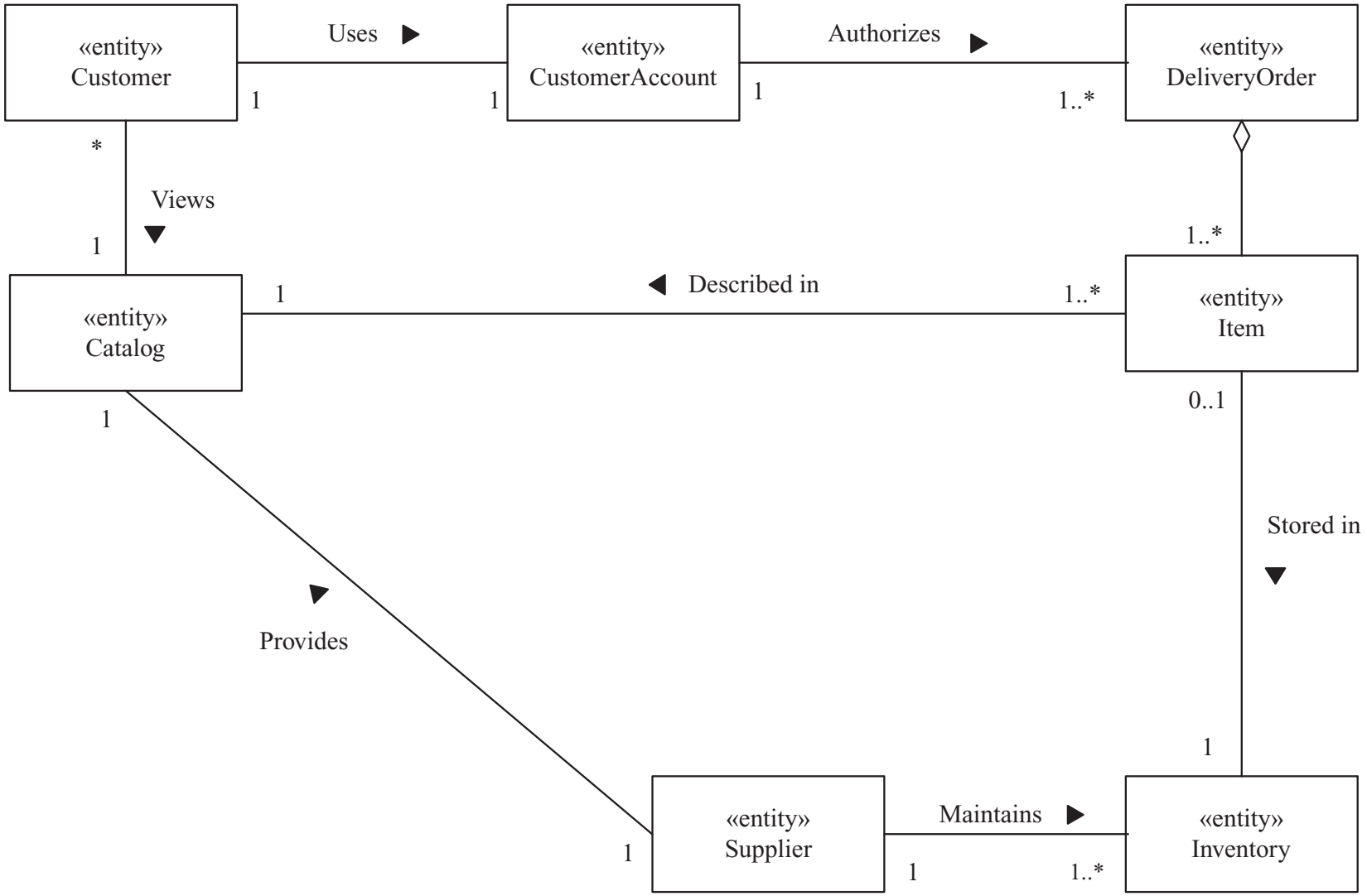
© UniRoma2 - Ingegneria del Software

Static Modeling

Software System Context Class Diagram

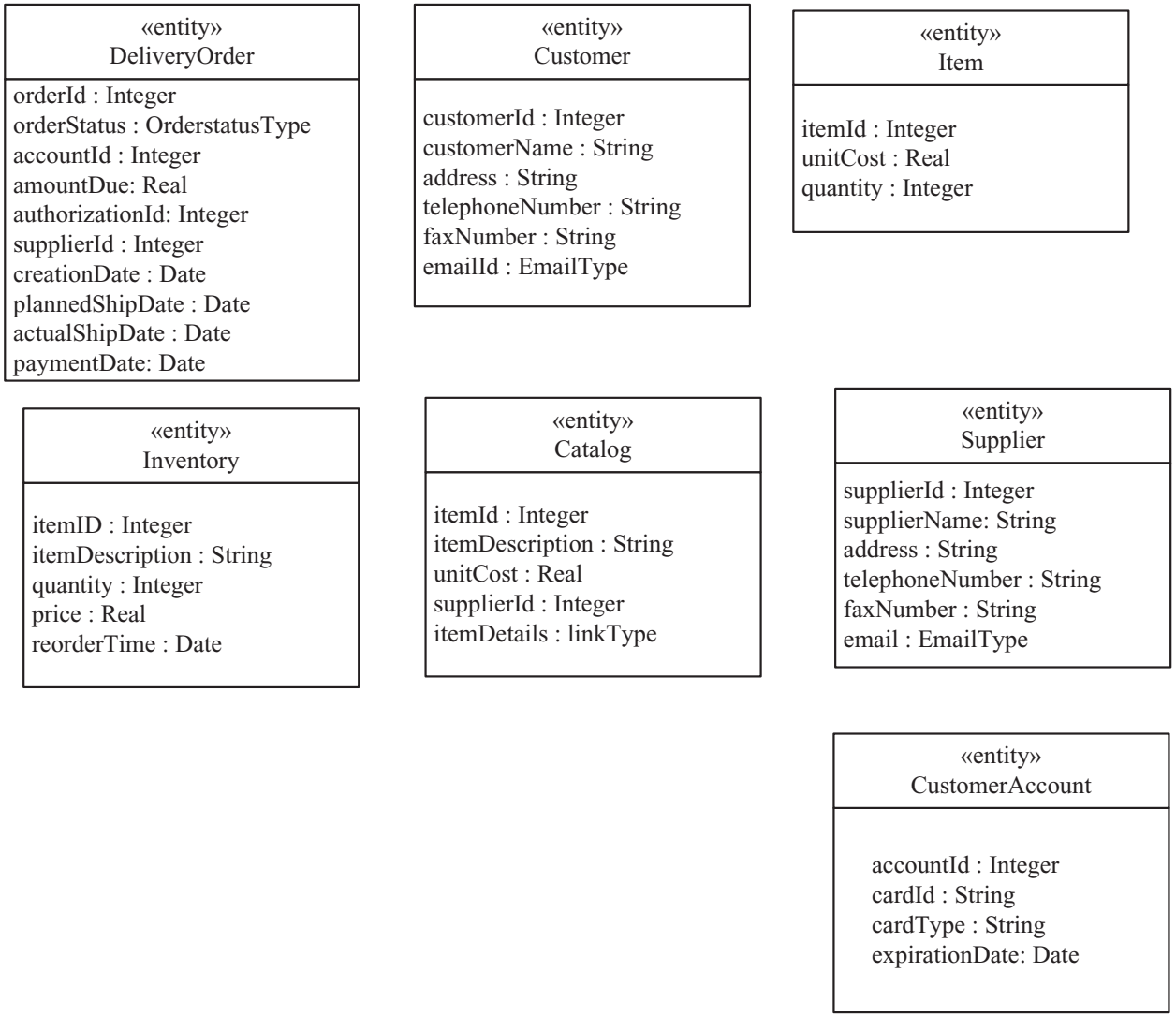


Entity Class Diagram



© UniRoma2 - Ingegneria del Software

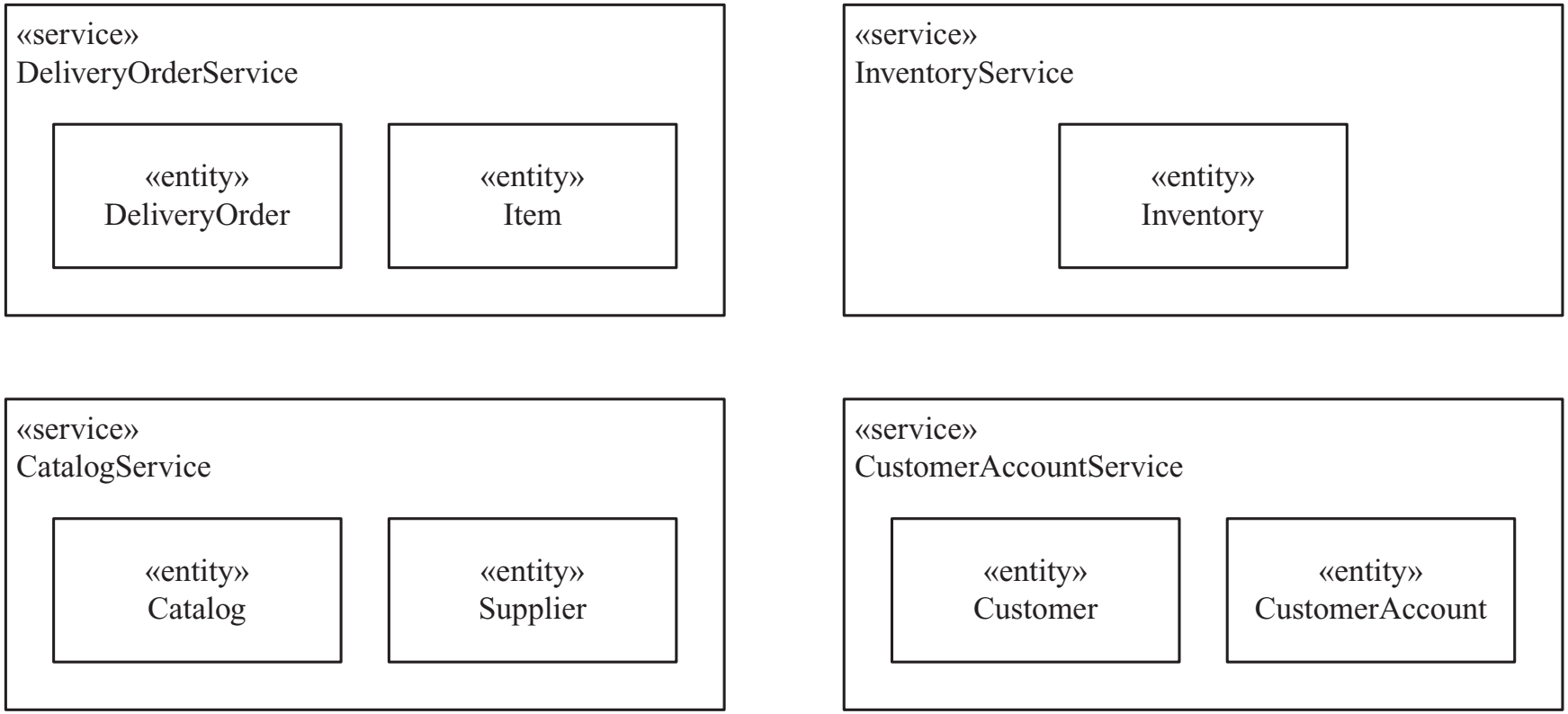
Entity Classes



Class Structuring

- The entity classes determined in the previous section are integrated into a service-oriented architecture by means of service classes
- Catalog Service, Customer Account Service, Delivery Order Service, and Inventory Service are service classes that provide access to the entity classes

Service and Entity Classes

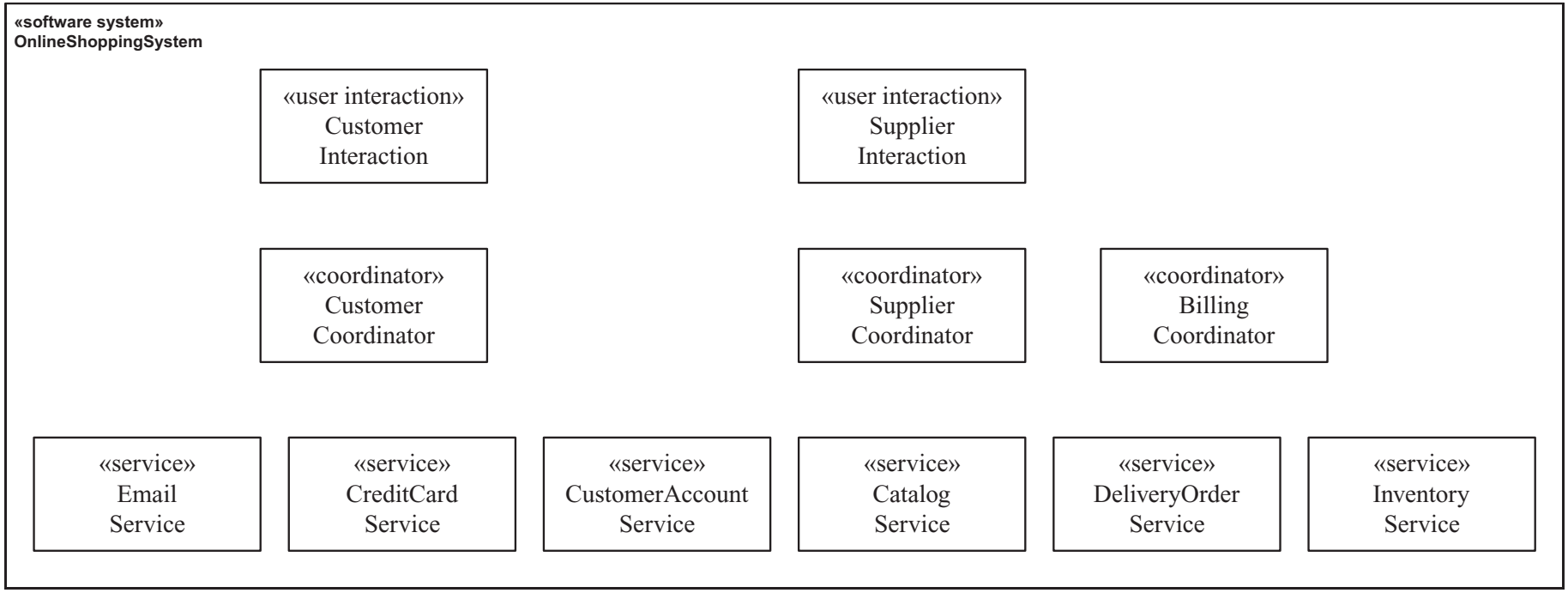


© UniRoma2 - Ingegneria del Software

Other classes

- There is also a *service class*, Credit Card Service, which deals with credit card authorization and charging
- Another *service class* is Email Service, which enables the Online Shopping System to send email messages to customers.
- *User interaction classes* are needed to interact with the external users – in particular, Customer Interaction and Supplier Interaction, which correspond to the actors in the use cases
- In addition, to coordinate and sequence the customer and supplier access to the online shopping services, two *coordinator classes*, Customer Coordinator and Supplier Coordinator, are provided,. A third autonomous coordinator, Billing Coordinator, is needed to deal with billing customers.

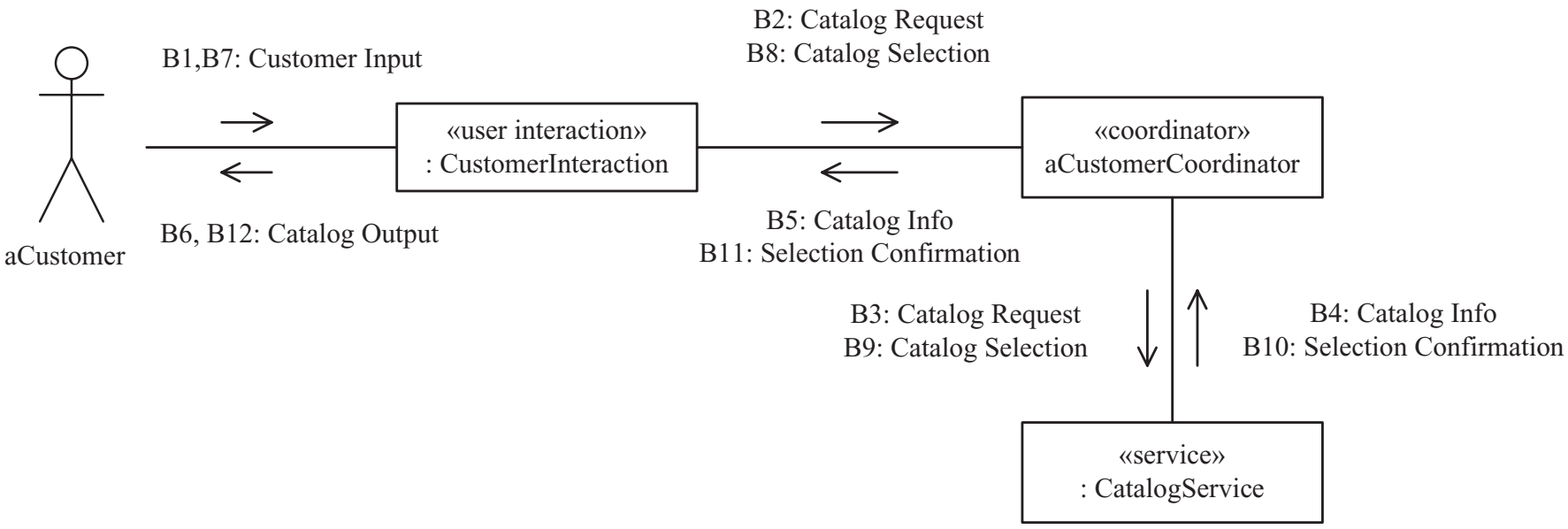
Class Structuring



© UniRoma2 - Ingegneria del Software

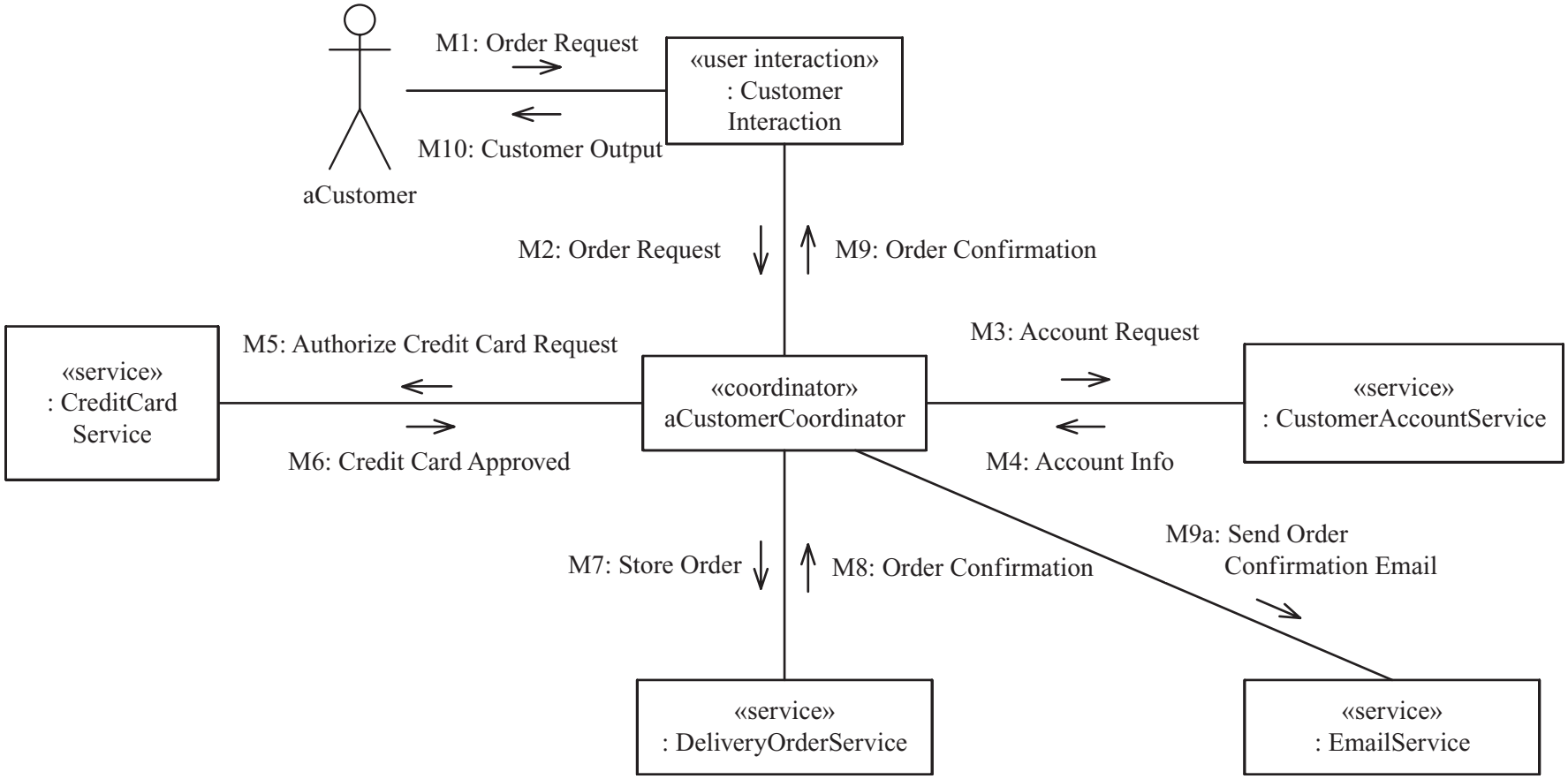
Dynamic Modeling

Communication diagram for the *Browse Catalog* UC

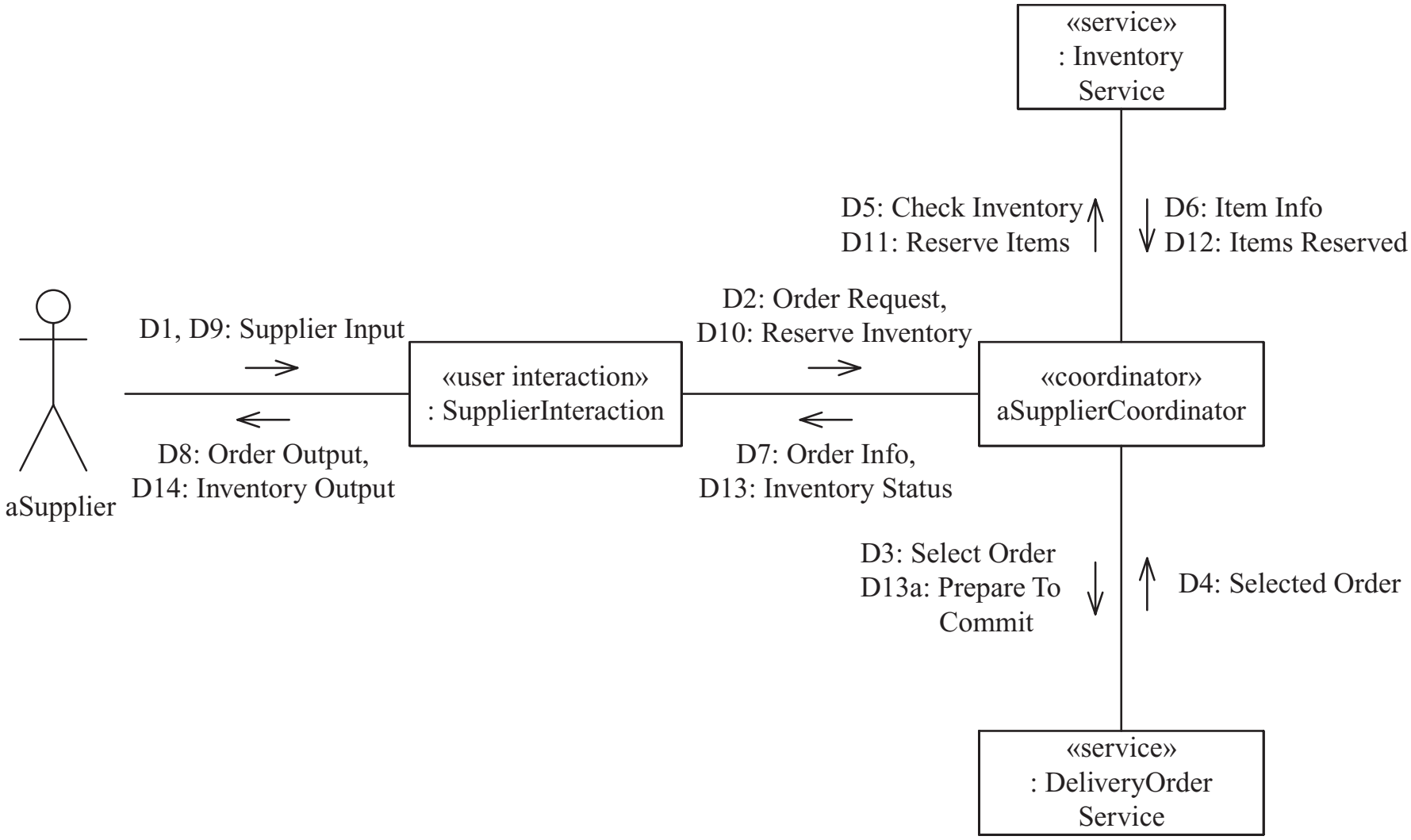


© UniRoma2 - Ingegneria del Software

Communication diagram for the *Make Order Request* UC

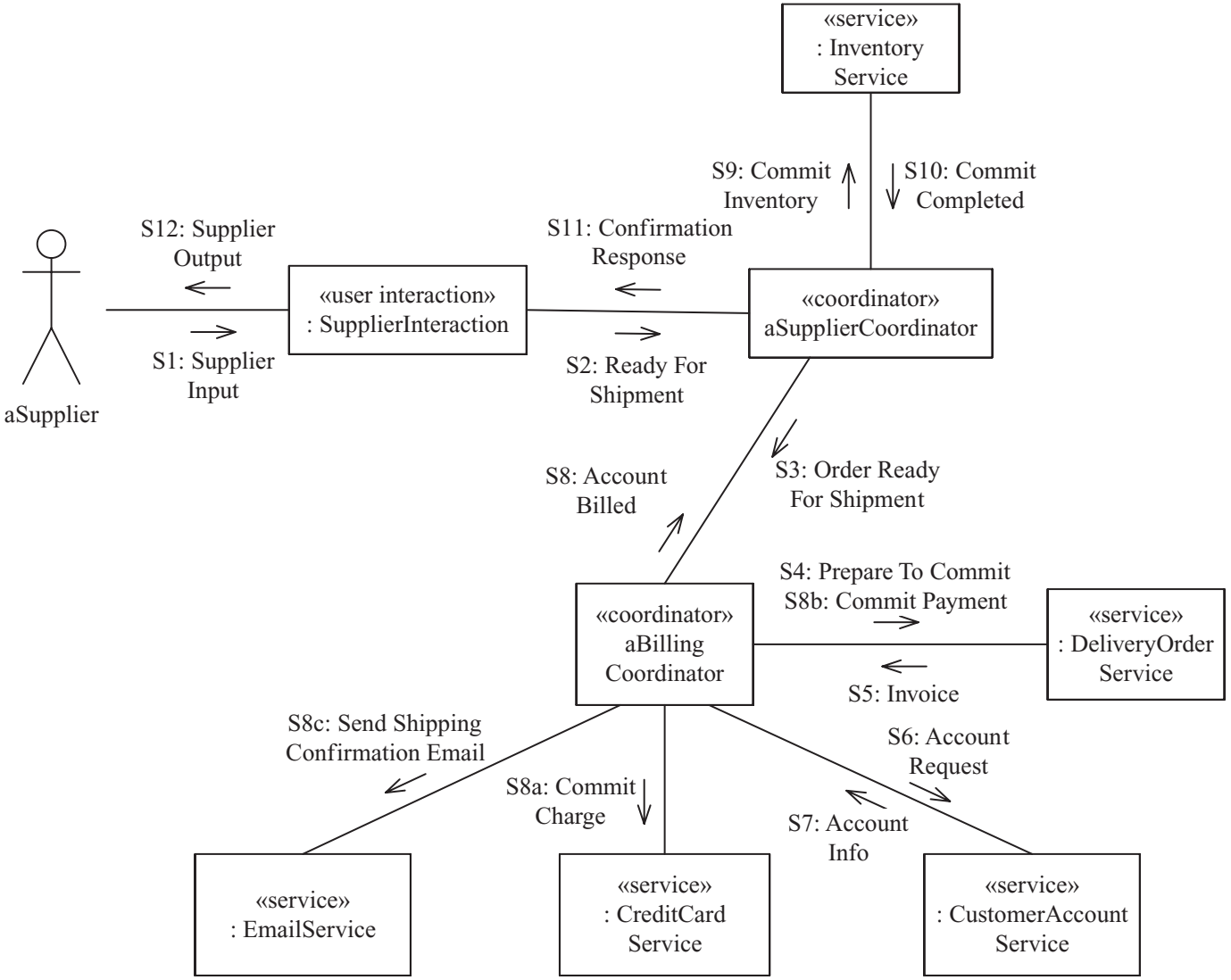


Communication diagram for the *Process Delivery Order* UC



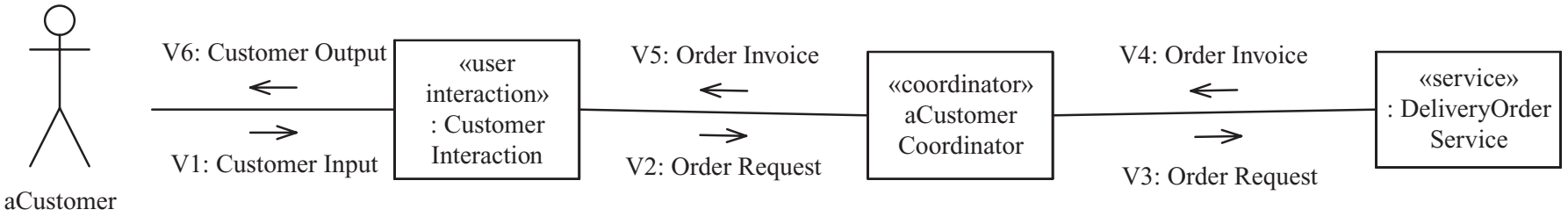
© UniRoma2 - Ingegneria del Software

Communication diagram for the *Confirm Shipment and Bill Customer* UC



© UniRoma2 - Ingegneria del Software

Communication diagram for the *View Order* UC



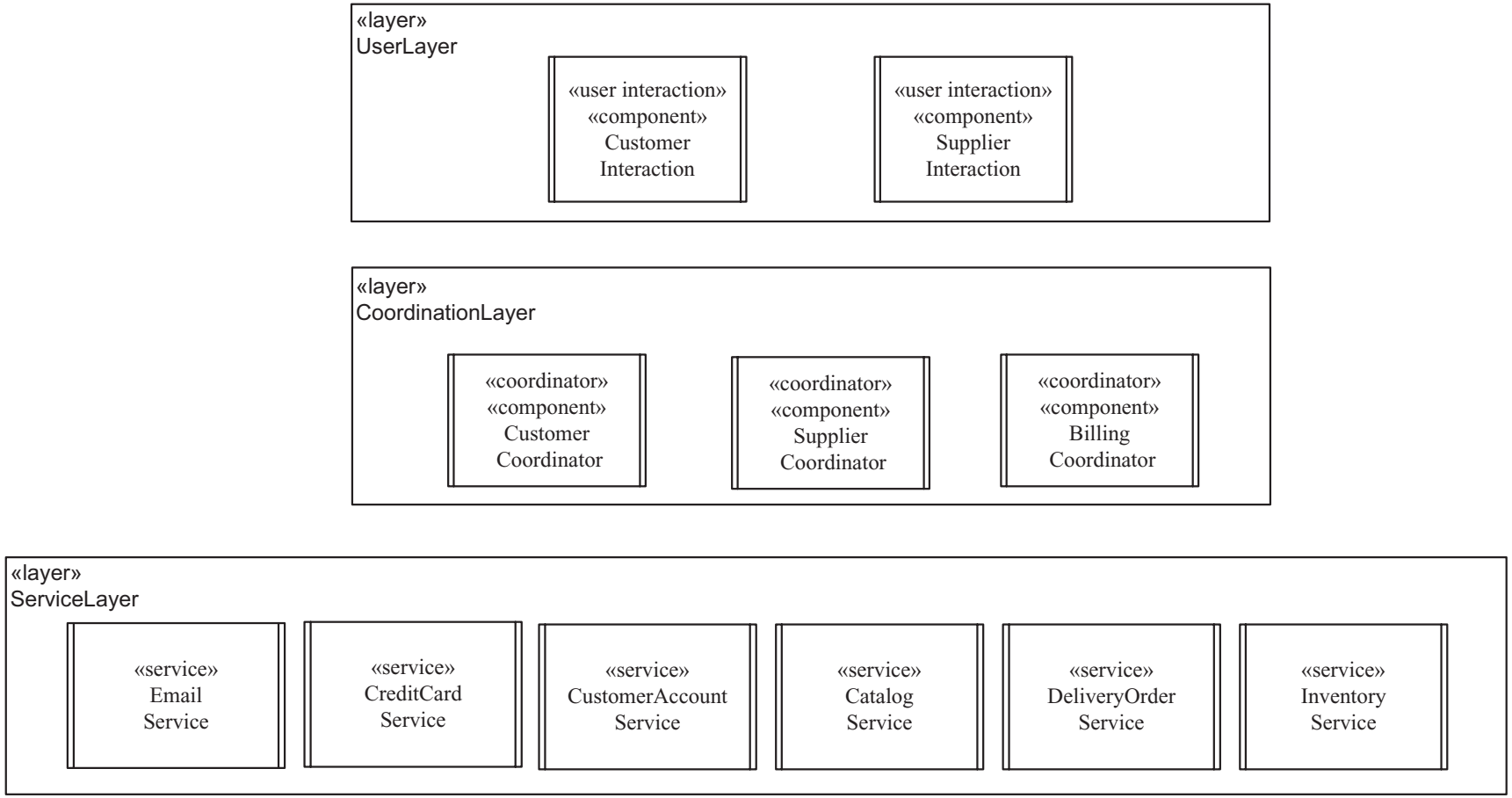
© UniRoma2 - Ingegneria del Software

Design Modeling

Design Modeling Overview

- The Online Shopping System is designed as a layered architecture based on the Layers of Abstraction architecture pattern
- The software architecture consists of three layers – a (data) service layer, a coordinator layer, and a user interaction layer
- Furthermore, because this system needs to be highly flexible and distributed, the decision is made to *design a service-oriented architecture*, in which distributed components can discover services and communicate with them
- Each component is depicted with the component stereotype (what kind of component it is, as specified by the component structuring criteria)
- The design of the component and service interfaces are determined by analysis of the communication diagrams for each use case

Layered Architecture



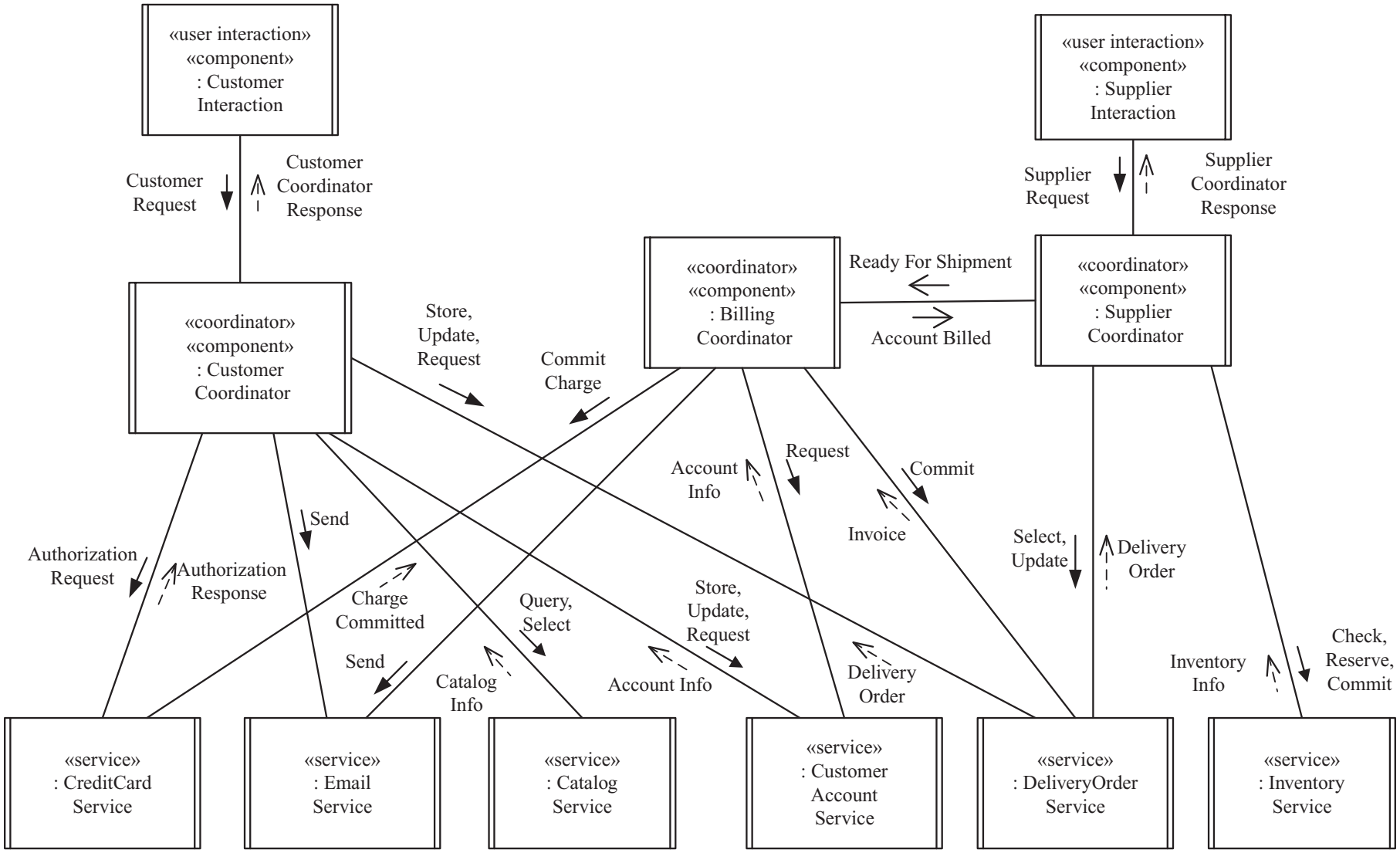
Architectural Communication Patterns

- Synchronous Message Communication with Reply
- Broker Handle
- Service Discovery
- Bidirectional Asynchronous Message Communication
- Two-Phase Commit

Concurrent Software Design

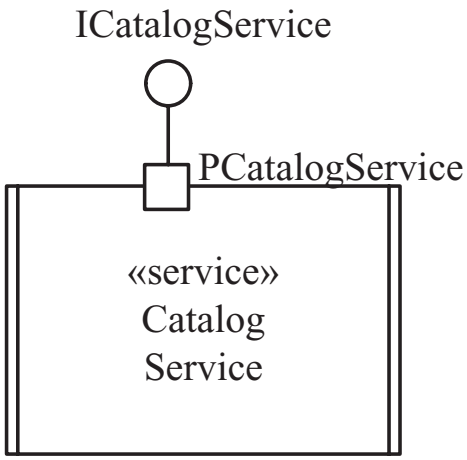
- To keep the design simple, the Synchronous Message Communication with Reply pattern has been widely used in this case study
- This approach has the disadvantage of suspending the client while it awaits a response from the service
- An alternative design to avoid suspending the client is to use the Asynchronous Message Communication with Callback pattern
- The Bidirectional Asynchronous Communication pattern is used for Supplier Coordinator and Billing Coordinator to communicate with each other in both directions

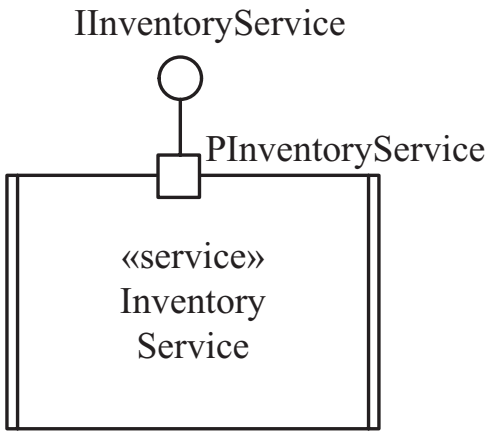
Concurrent Communication Diagram

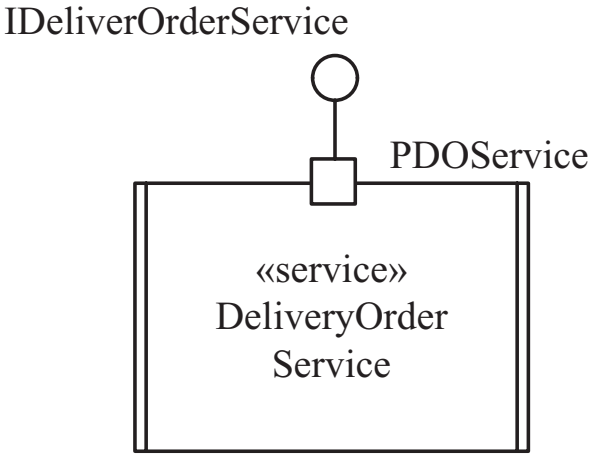


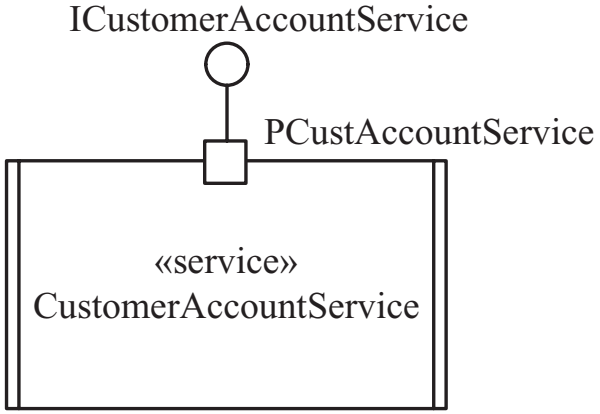
© UniRoma2 - Ingegneria del Software

Component Ports and Interfaces for Services



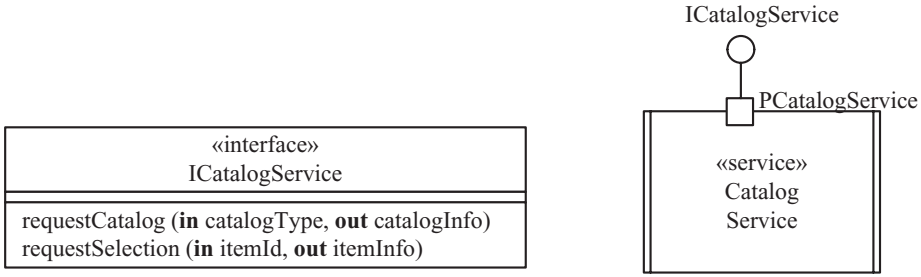


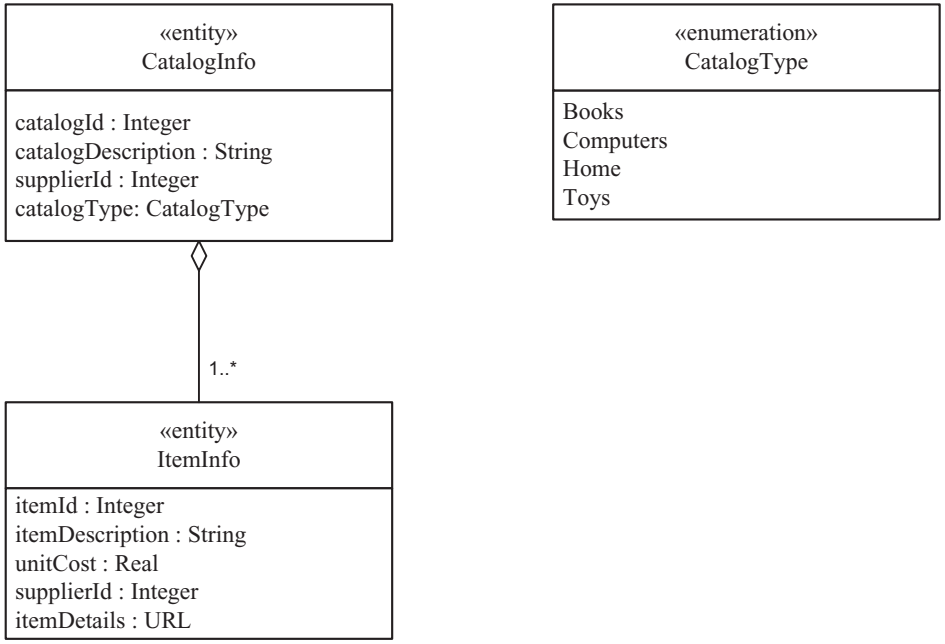




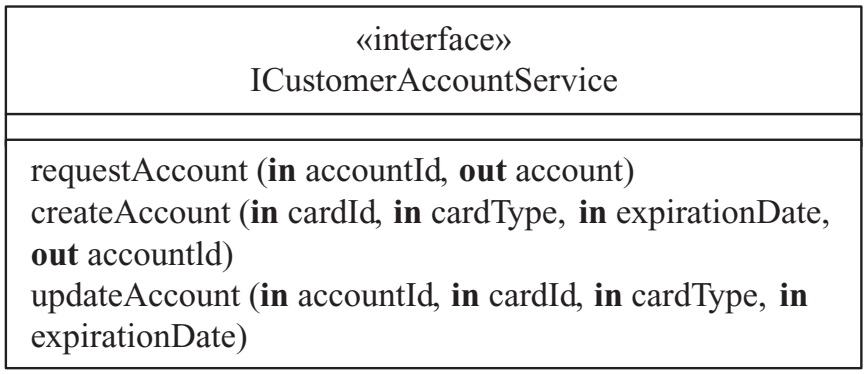
© UniRoma2 - Ingegneria del Software

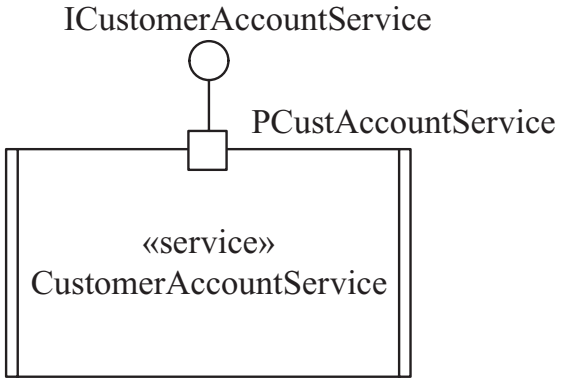
Service Interface for *Catalog* Service

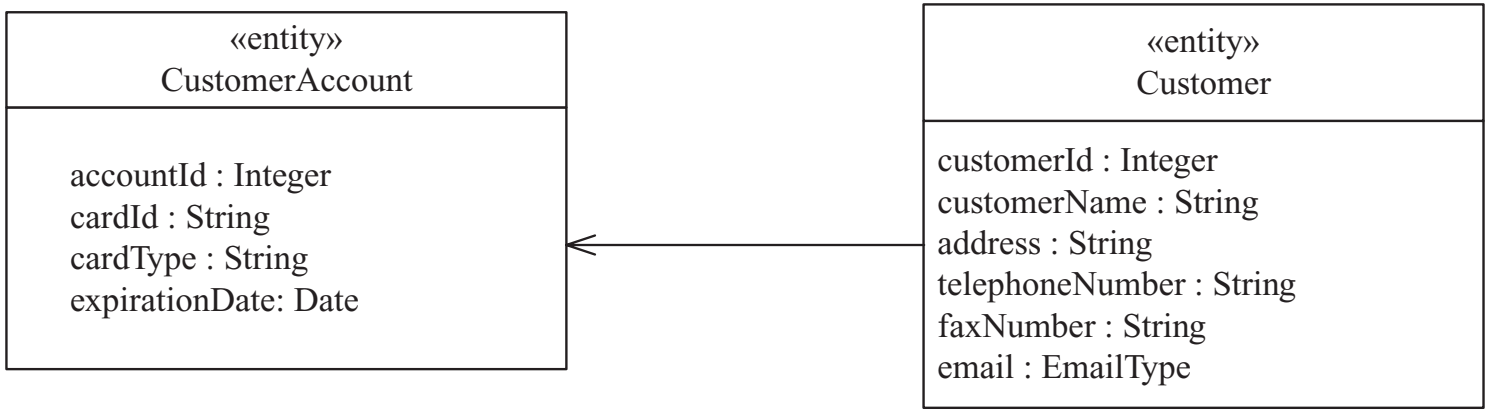




Service Interface for *Customer Account* Service

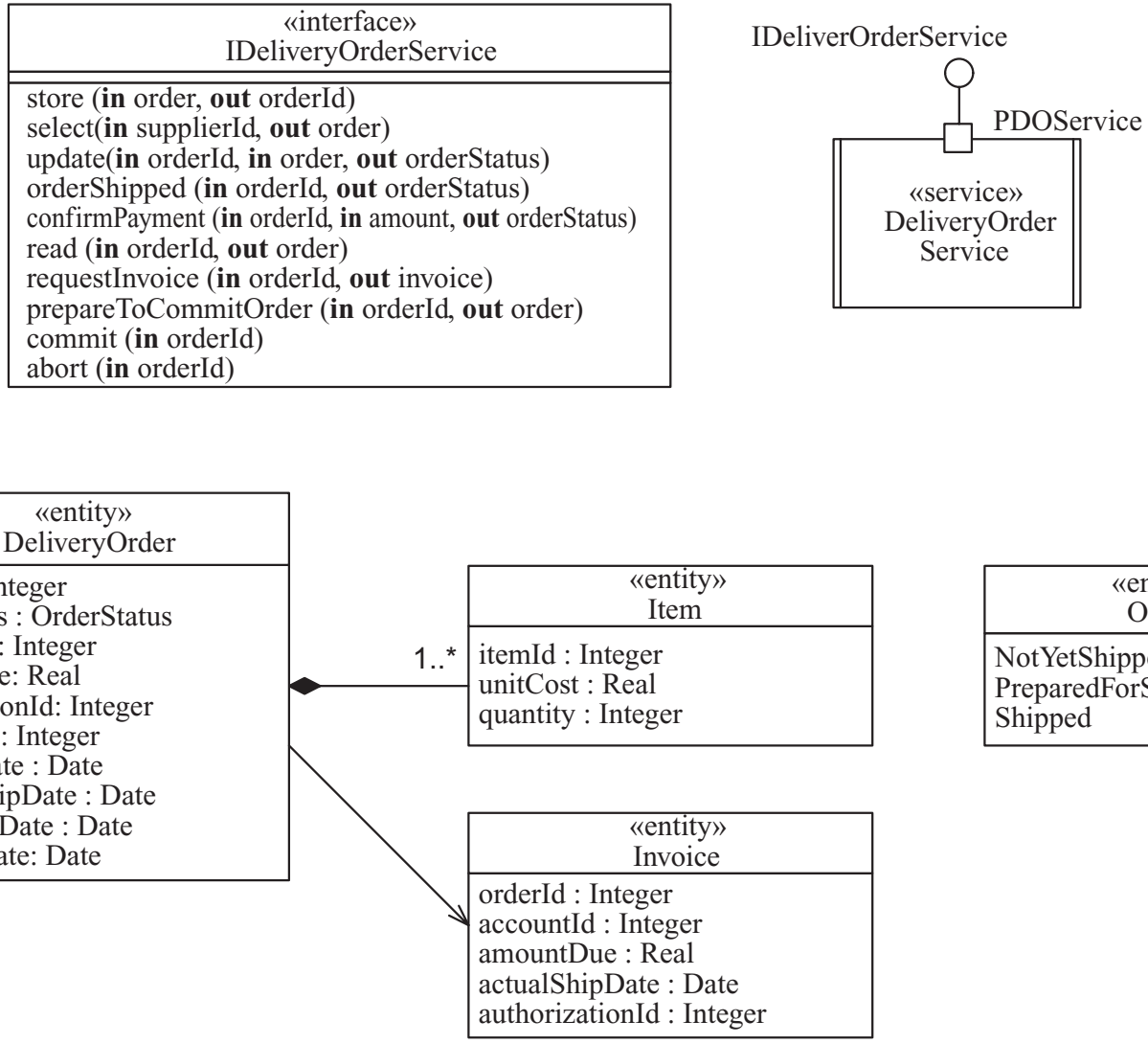






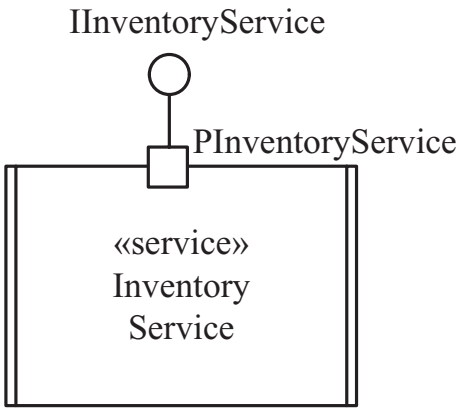
© UniRoma2 - Ingegneria del Software

Service Interface for *Delivery Order* Service



© UniRoma2 - Ingegneria del Software

Service Interface for *Inventory* Service



«interface»
IInventoryService

checkInventory (**in** itemId, **out** inventoryStatus)
update (**in** itemId, **in** amount)
reserveInventory (**in** itemId, **in** amount)
commitInventory (**in** itemId, **in** amount)
abortInventory (**in** itemId, **in** amount)

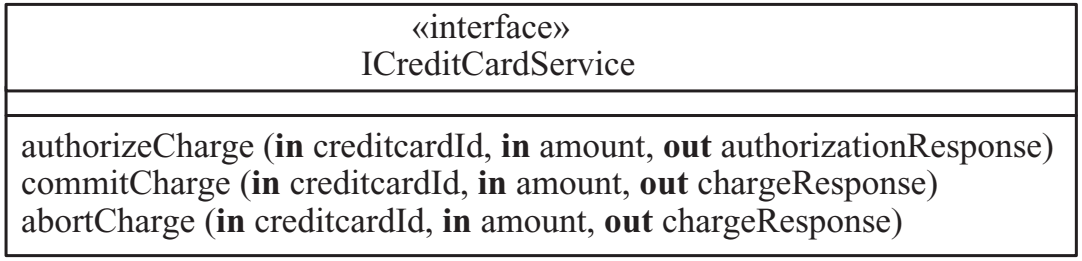
«entity»
Inventory

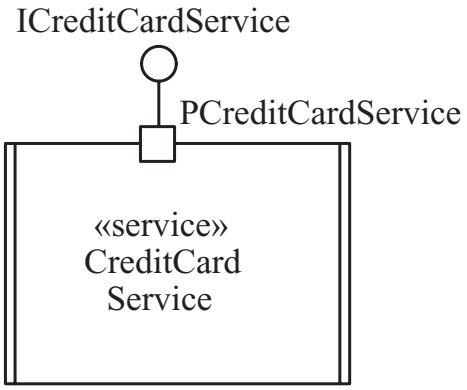
itemID : Integer
itemDescription : String
quantity : Integer
quantityReserved: Integer
price : Real
reorderTime : Date

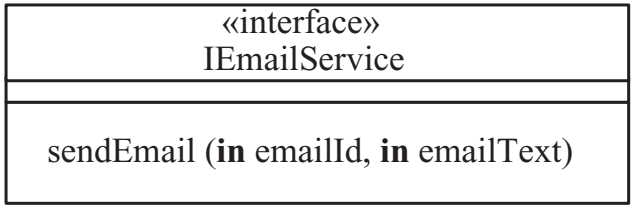
«entity»
InventoryStatus

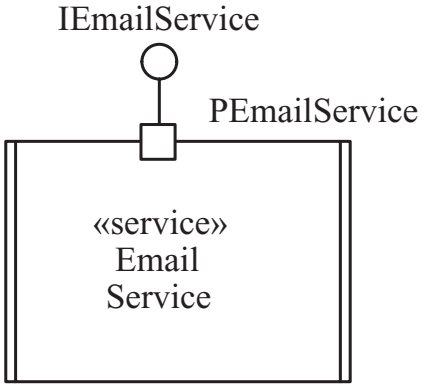
itemID : Integer
currentAmount: Integer
quantityAfterShipped: Integer
reorderTime : Date

Service Interface for *Credit Card* and *Email* services

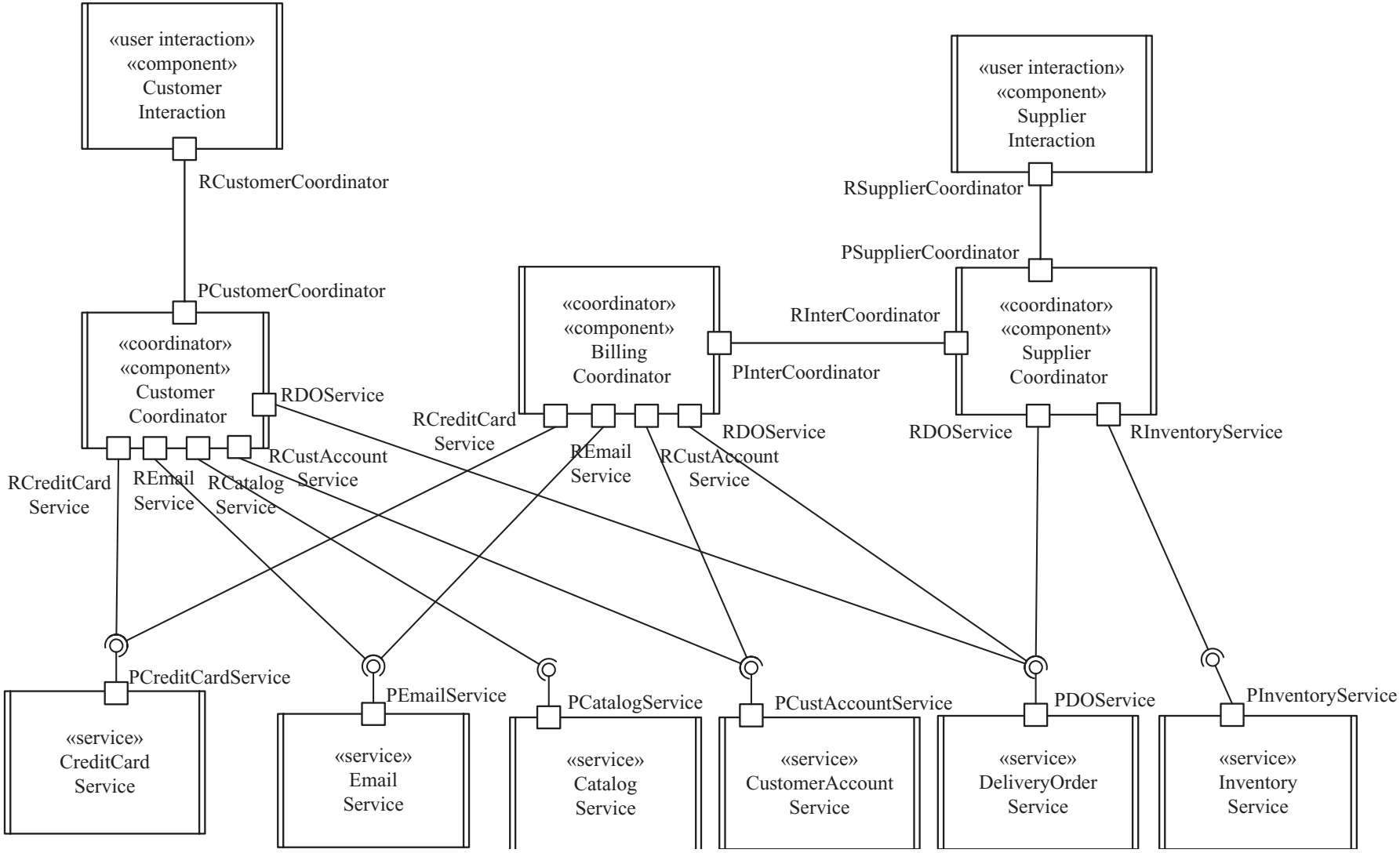




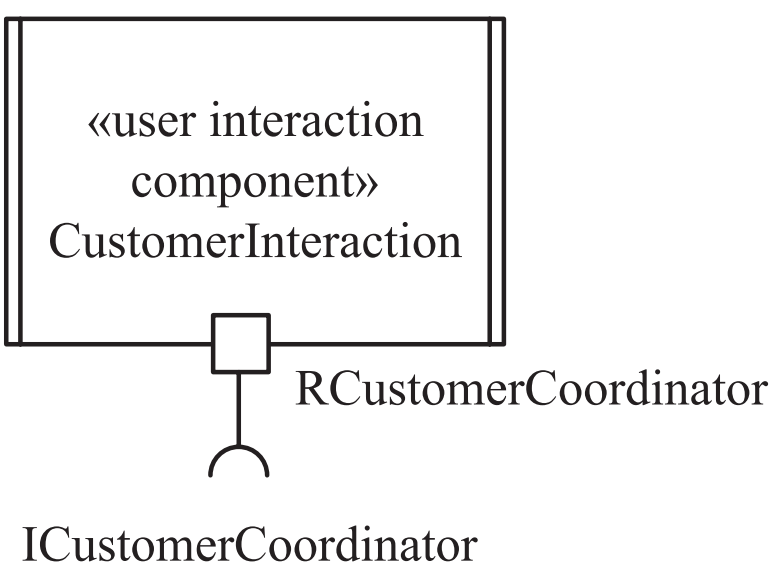


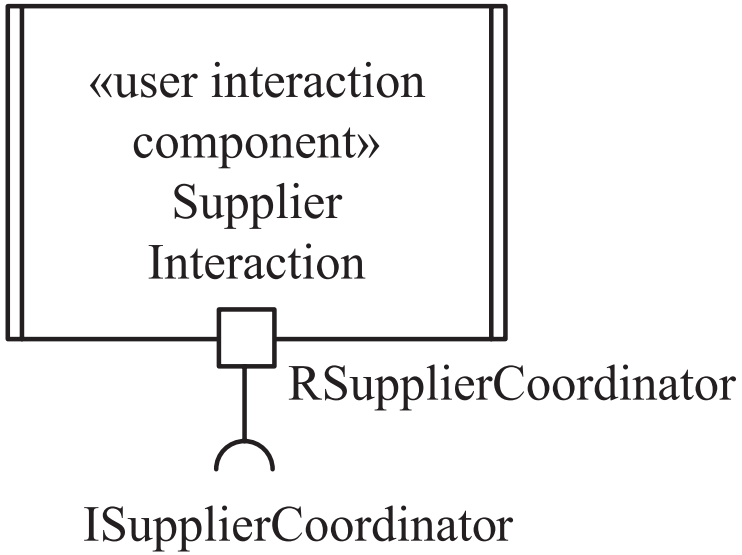


Service-Oriented Software Architecture



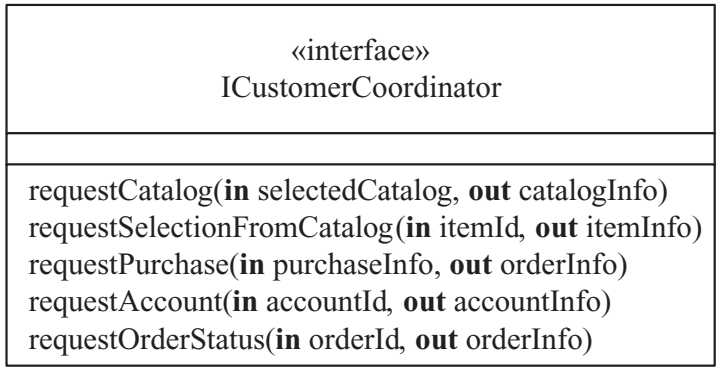
Component ports and interfaces for *Customer Interaction* and *Supplier Interaction*

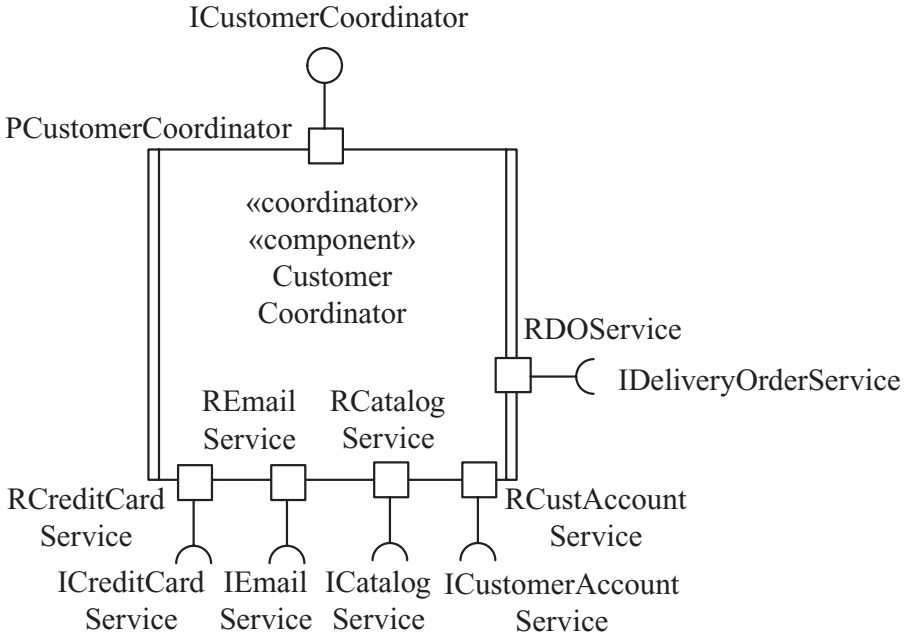




© UniRoma2 - Ingegneria del Software

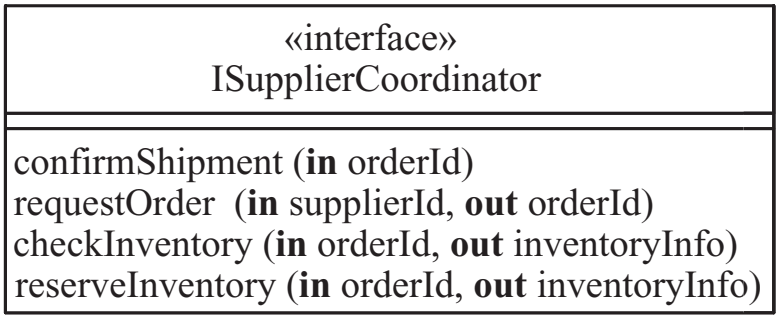
Component ports and interfaces for *Customer* Coordinator

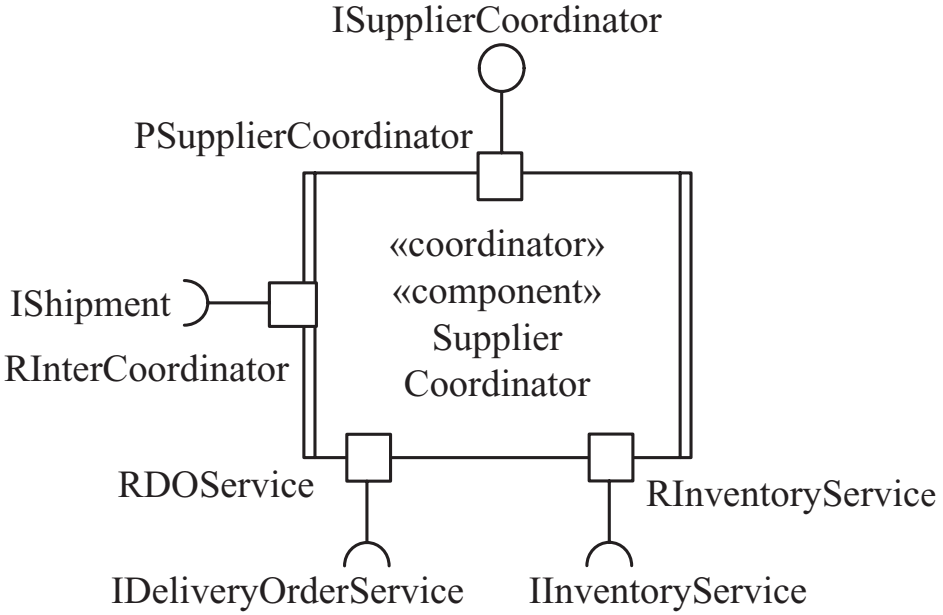




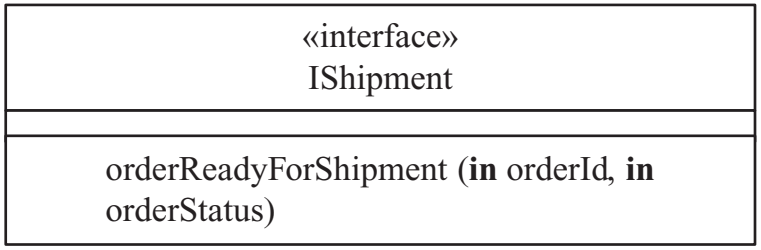
© UniRoma2 - Ingegneria del Software

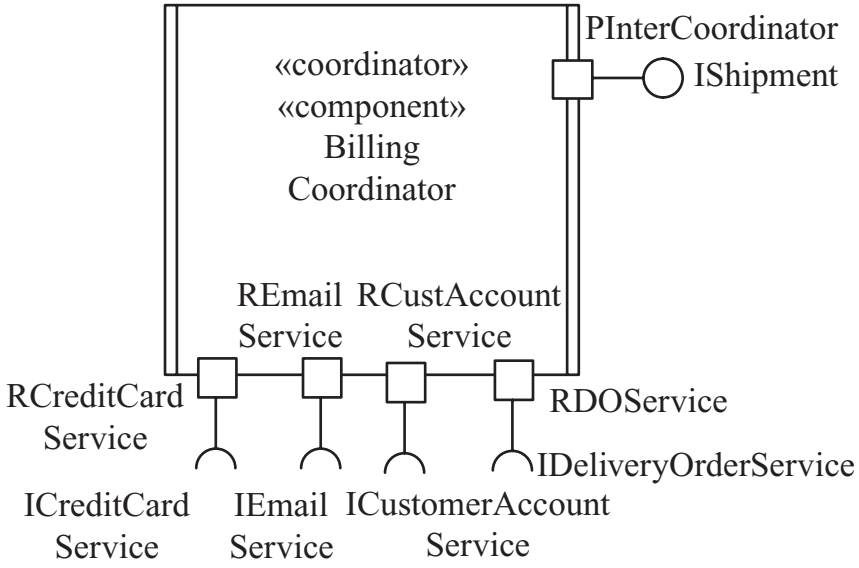
Component ports and interfaces for *Supplier* Coordinator





Component ports and interfaces for *Billing* Coordinator





Service Reuse

- With the SOA paradigm, once the services have been designed and their interfaces specified, the service interface information can be registered with a service broker
- Services can be composed into new applications
- This case study has described an Online Shopping System
- However, other electronic commerce systems could be designed that would reuse the services provided by the Online Shopping System, such as Catalog Service, Delivery Order Service, and Inventory Service