The Unified Modeling Language

An Introduction

Agenda

- Background
- Things
- Relationships
- Summary

Background

- Pictures are Nice
- The Unified Movement
  - Three Amigos
    - Booch, Rumbaugh, Jacobson
- UML is a melding of other Notations
  - A Graphical Language, not a Methodology
  - Complex
  - Just use what you need

Things

- Parts/Views of a System
- Three Categories of Diagrams
  - Structural
  - Behavioral
  - Grouping

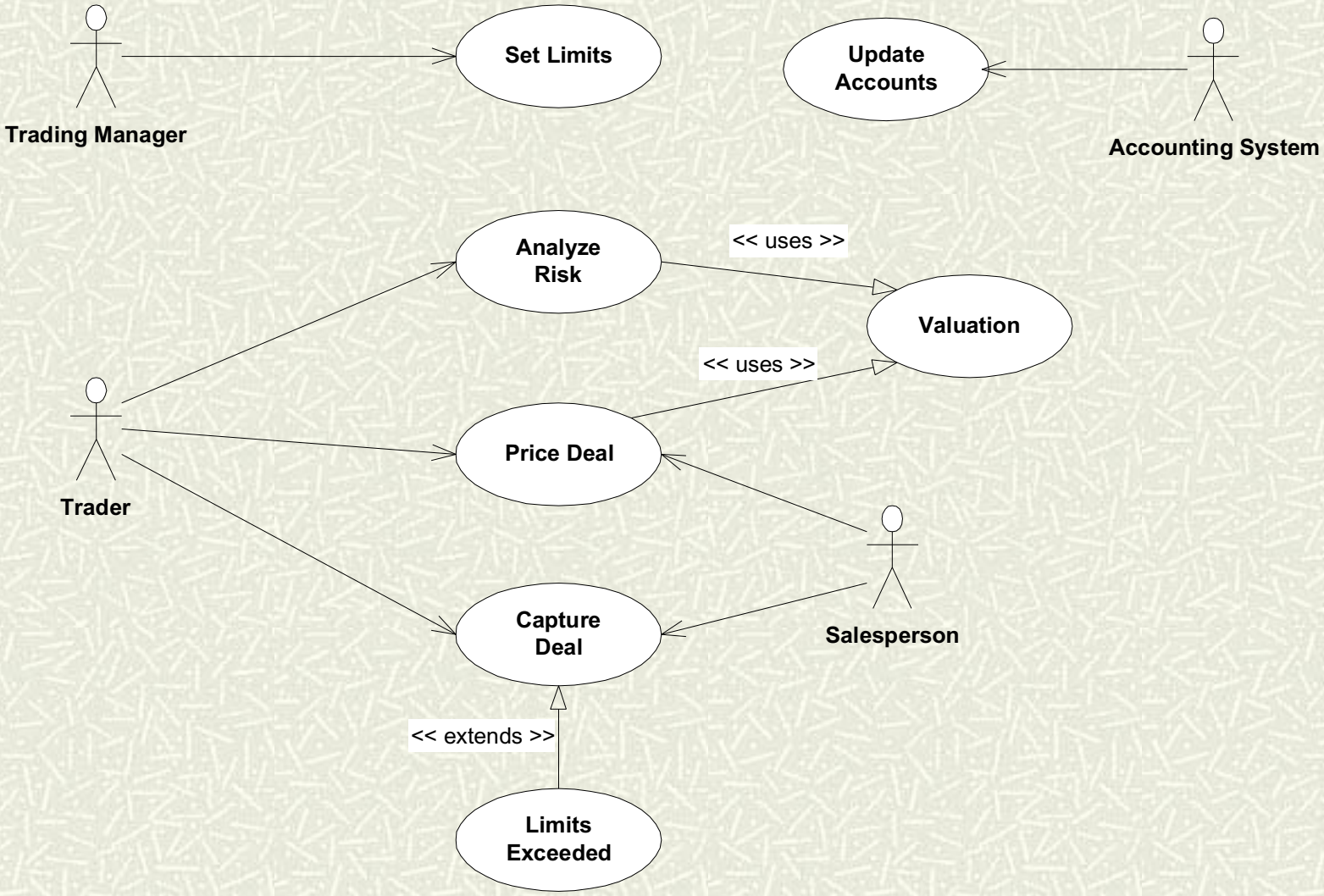
Structural Diagrams

- Use Case
- Class
- Interface
- Component
- Collaboration

Use Cases

- Documents a user's interaction with the system
  - An external view
- Often just recorded in text
- Diagrams can get very complex

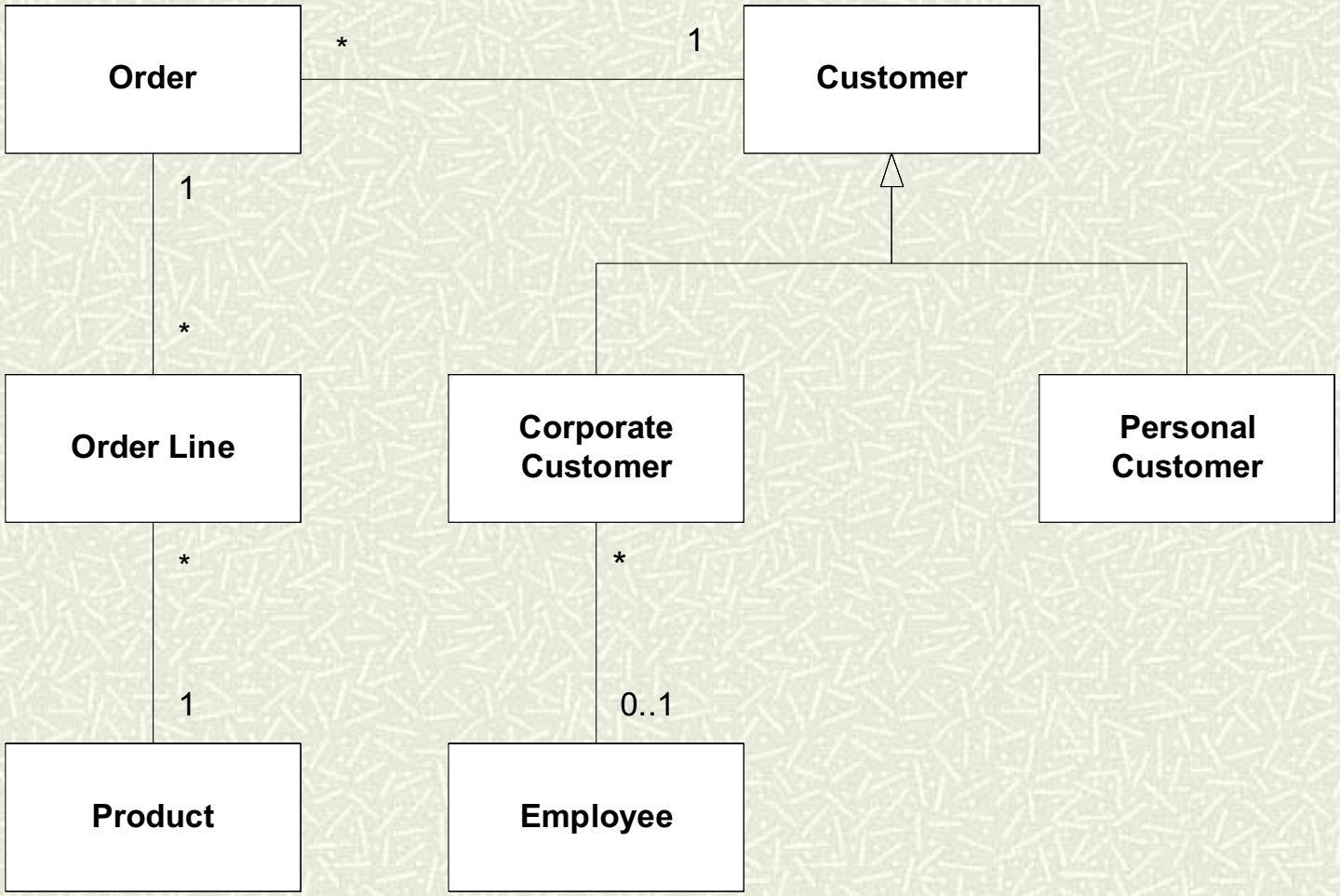
Use Case Diagram



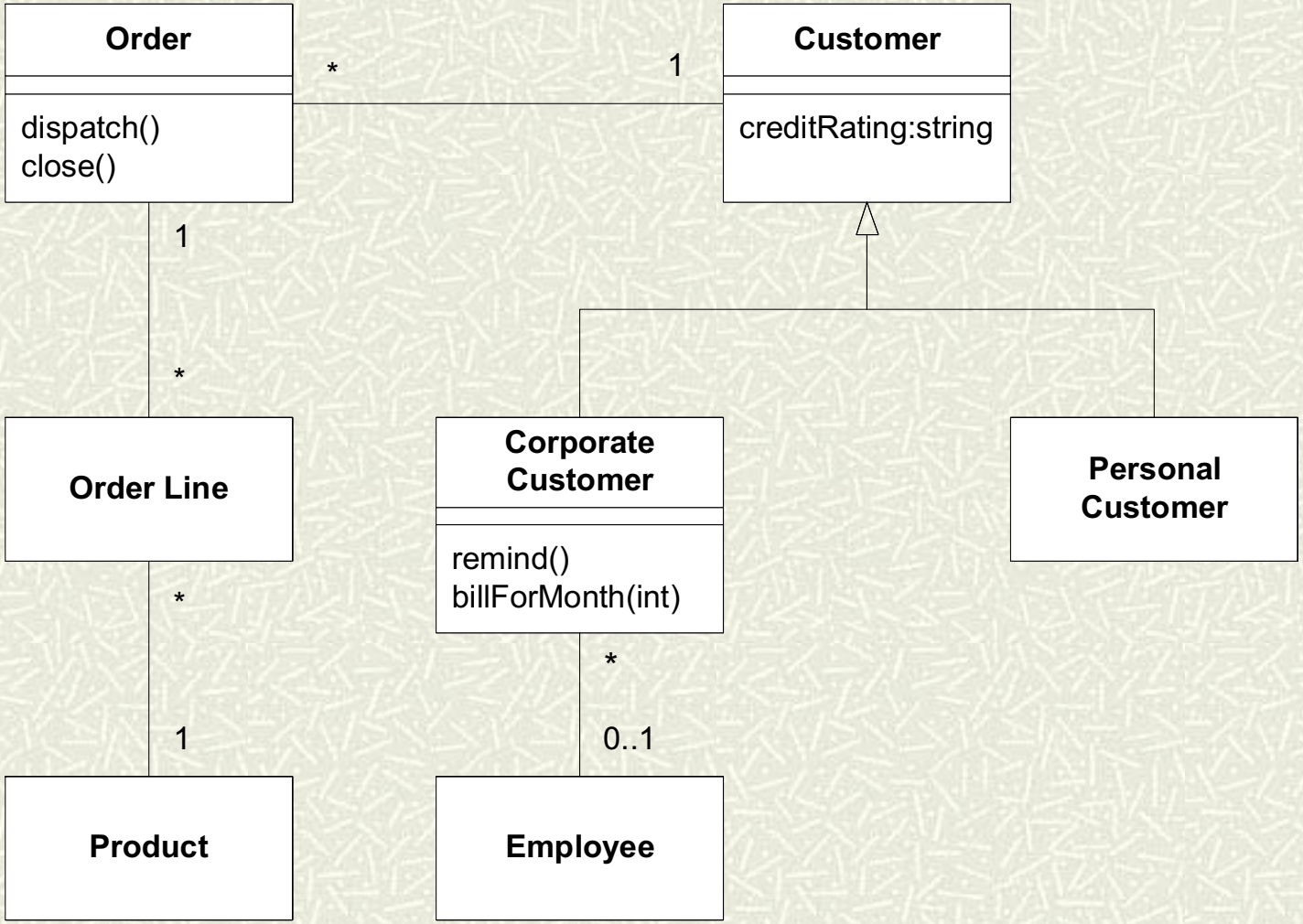
Class Diagrams

- Denotes key system abstractions and their static relationships
  - associations
  - subtypes
- Crucial for an overall “picture” of the system
- Can capture different levels of granularity
  - views with increasing detail
    - Conceptual (Domain View)
    - Specification (Class Interfaces)
    - Implementation (Hidden Attributes & Behavior)
- Often the starting point for implementation

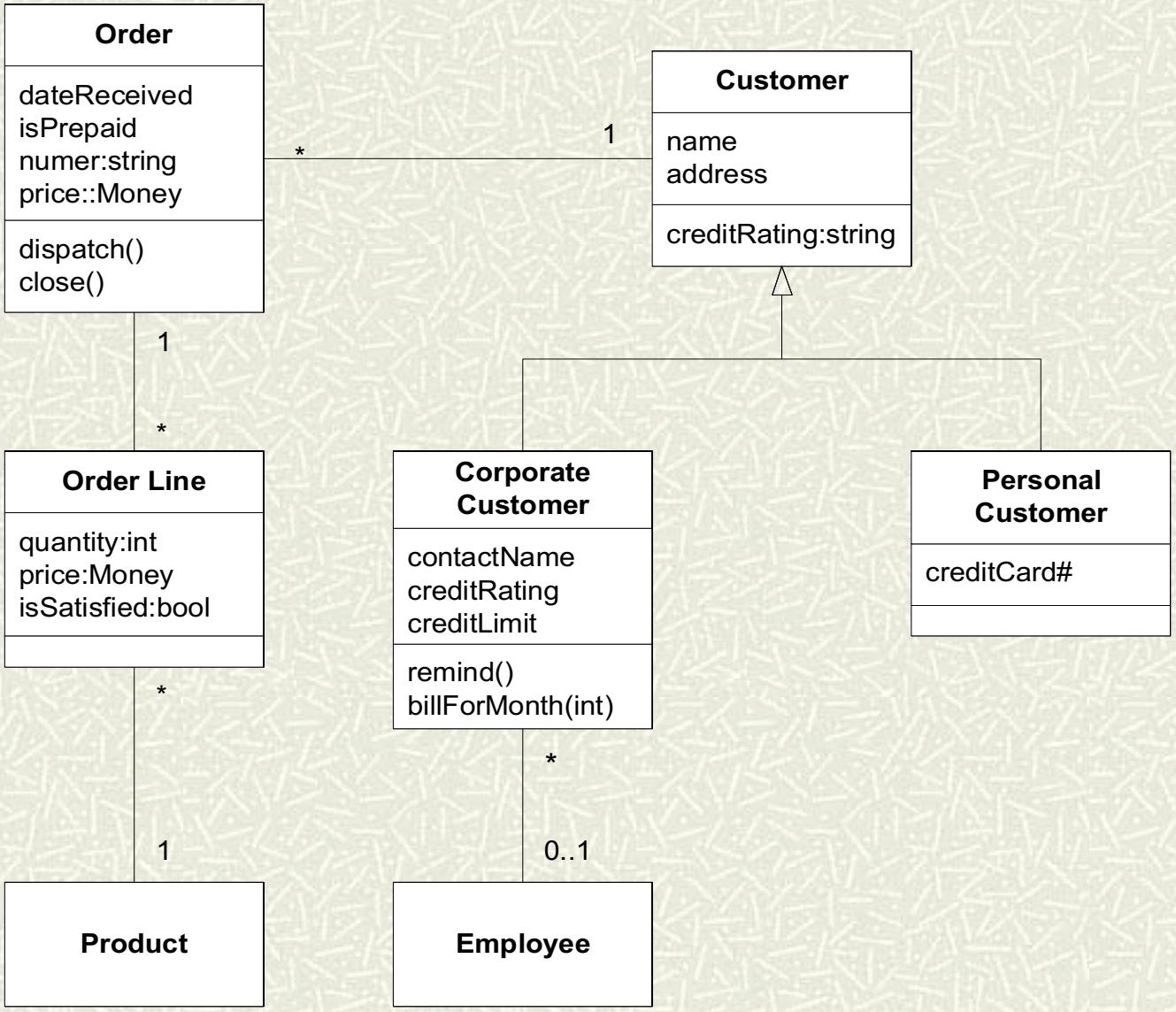
Conceptual Class Diagram



Specification Class Diagram



Implementation Class Diagram



Interfaces

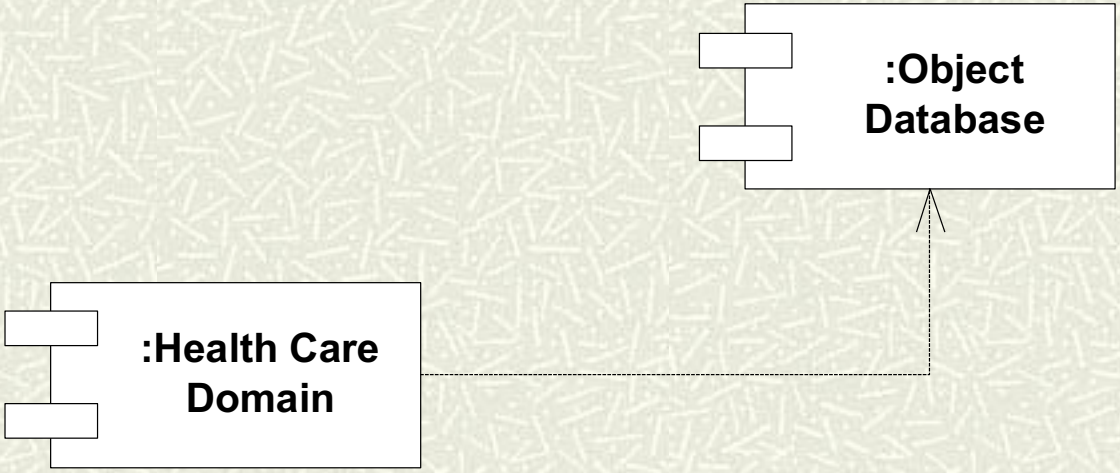
- Define external behavior
  - A set of methods
- Key to Distributed Programming
  - COM
  - CORBA
  - Java RMI

Interface Diagram

Component Diagrams

- A Component is an artifact with a well-defined interface boundary
  - like a library, or a COM component

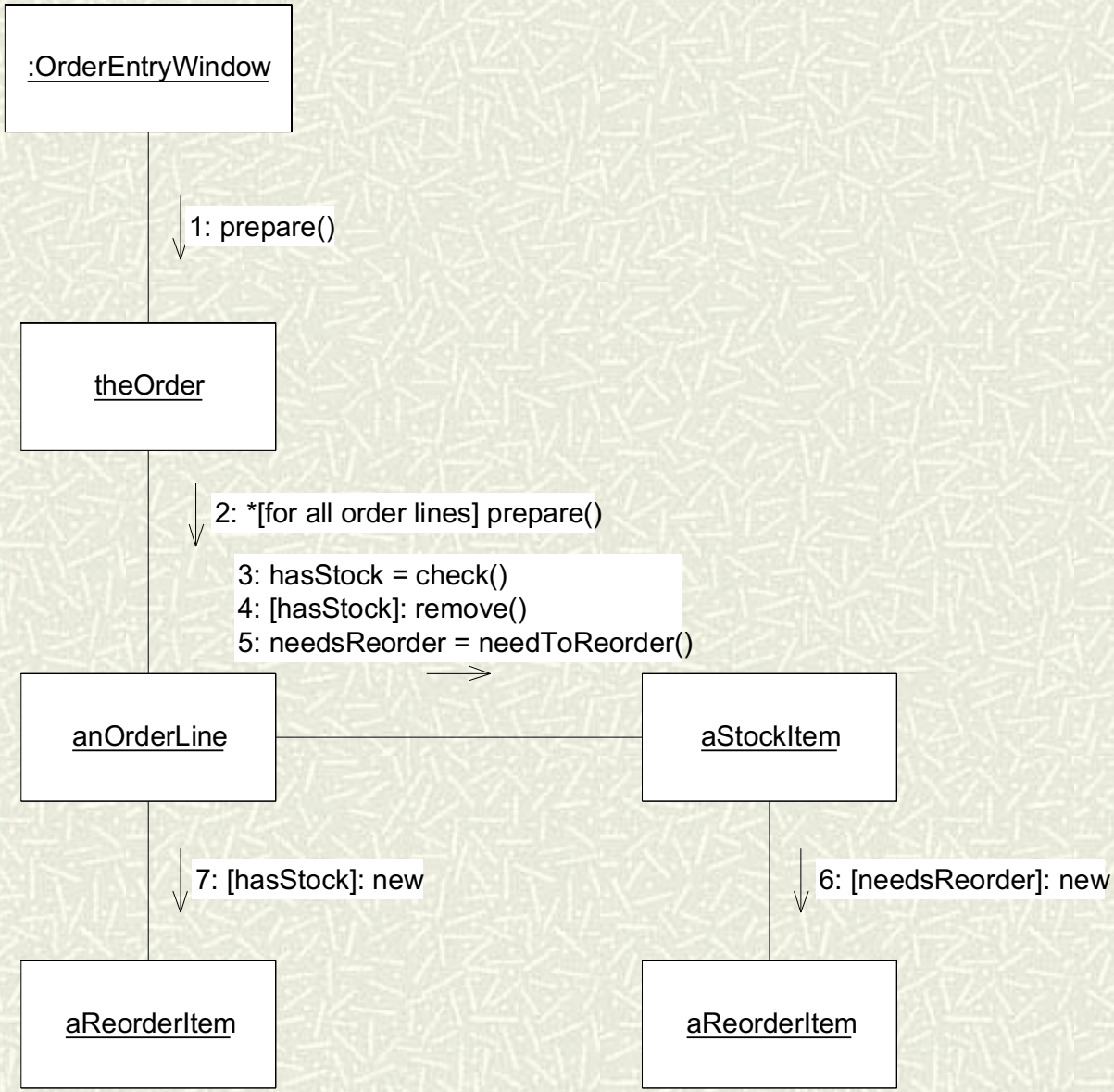
Component Diagram



Collaboration Diagrams

- An Interaction Diagram
- Shows Links between objects
- Sequence shown by numbering

Collaboration Diagram



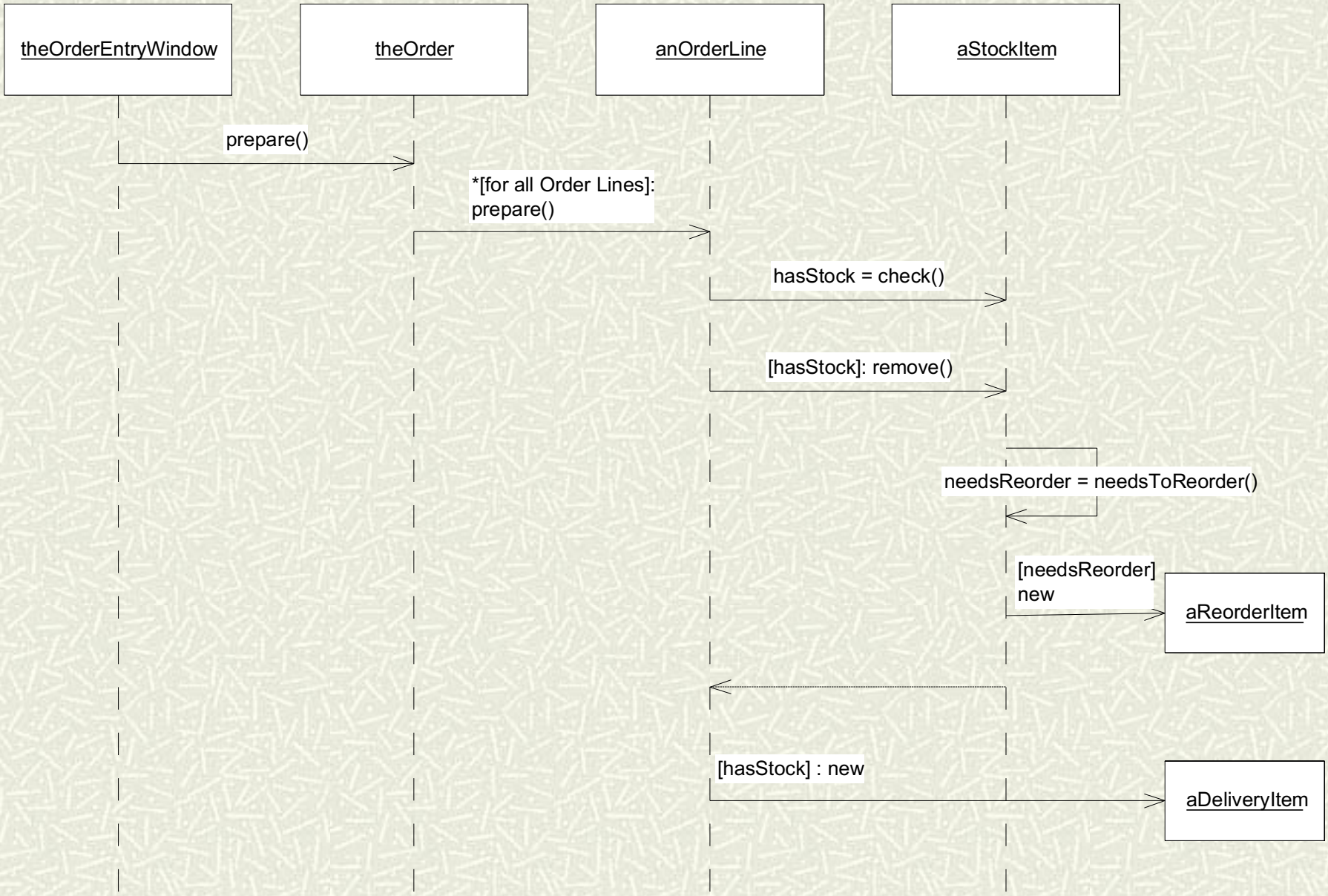
Behavioral Diagrams

- Sequence
- State Machine
- Activity

Sequence Diagrams

- Similar to Collaboration Diagrams
- Sequence visible via object lifeline

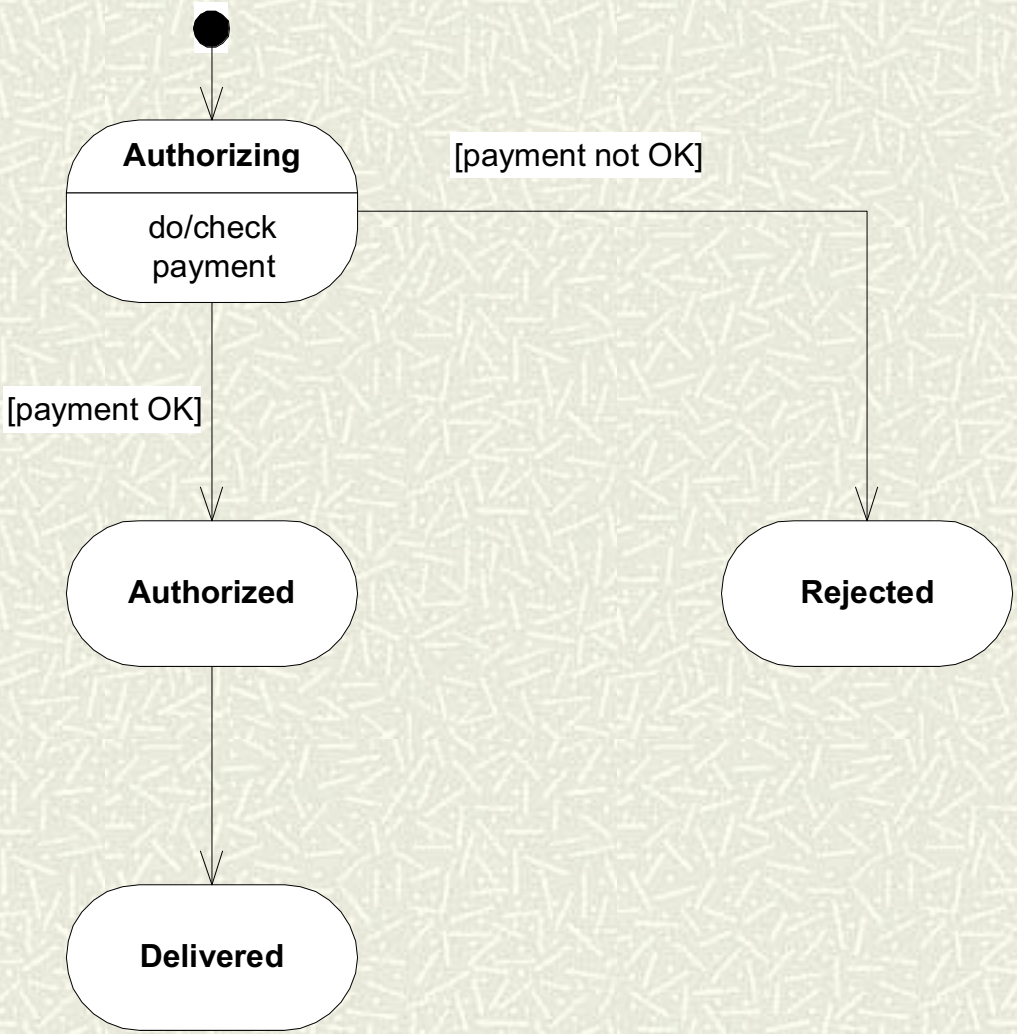
Sequence Diagram



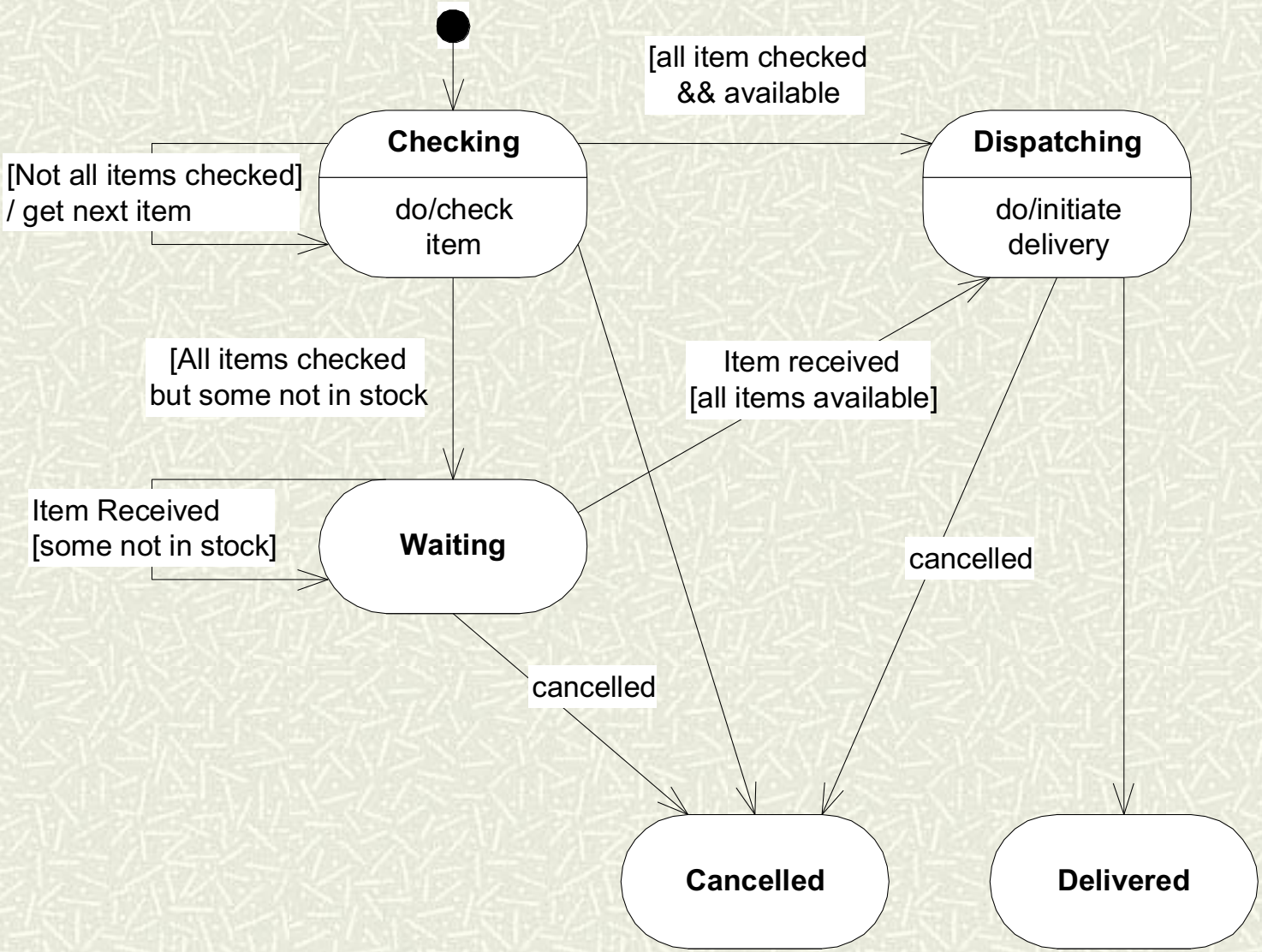
State Diagrams

- Classical State Machine Diagram
- Records complicated interactions governed by state
- Can show concurrency

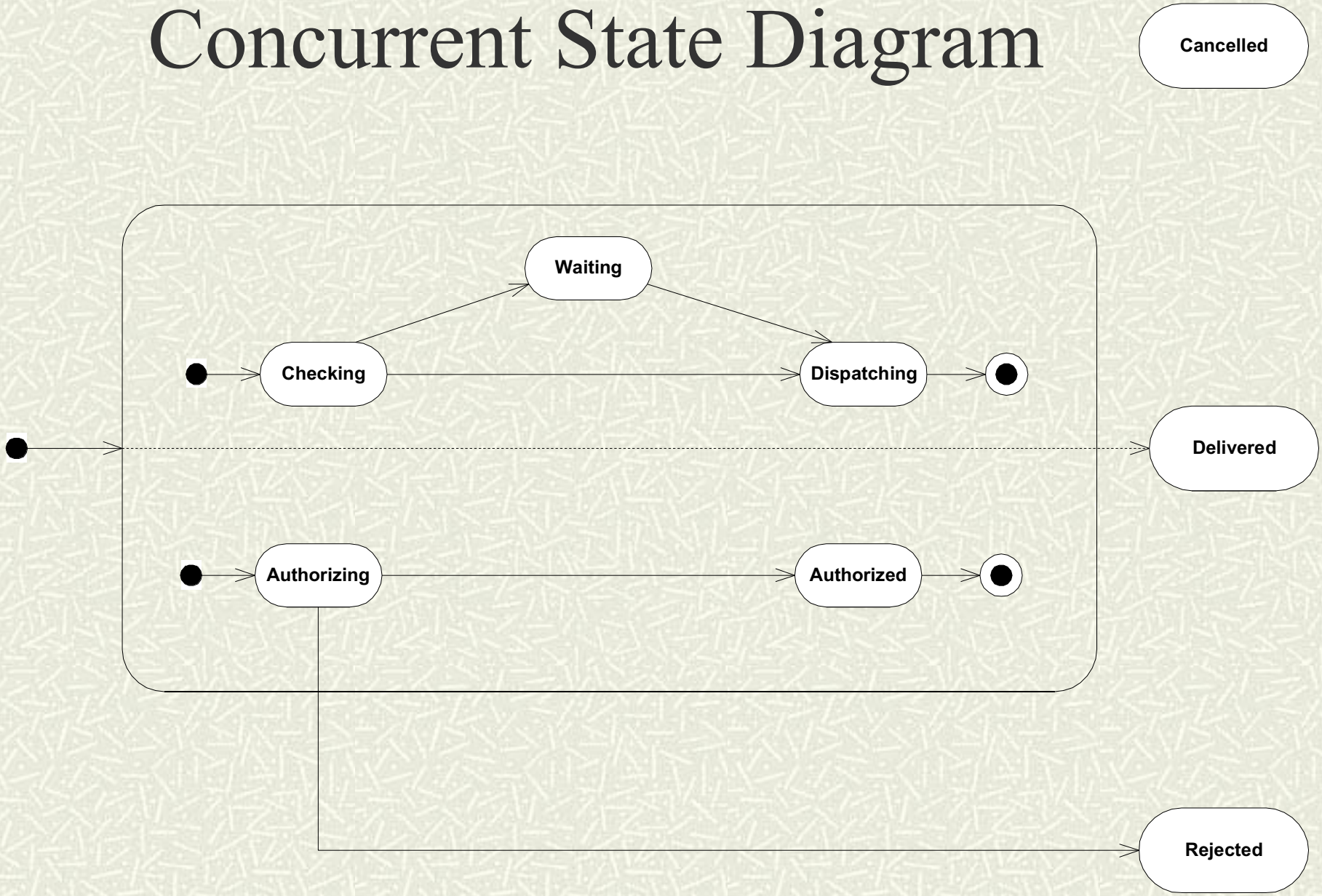
State Diagram



State Diagram

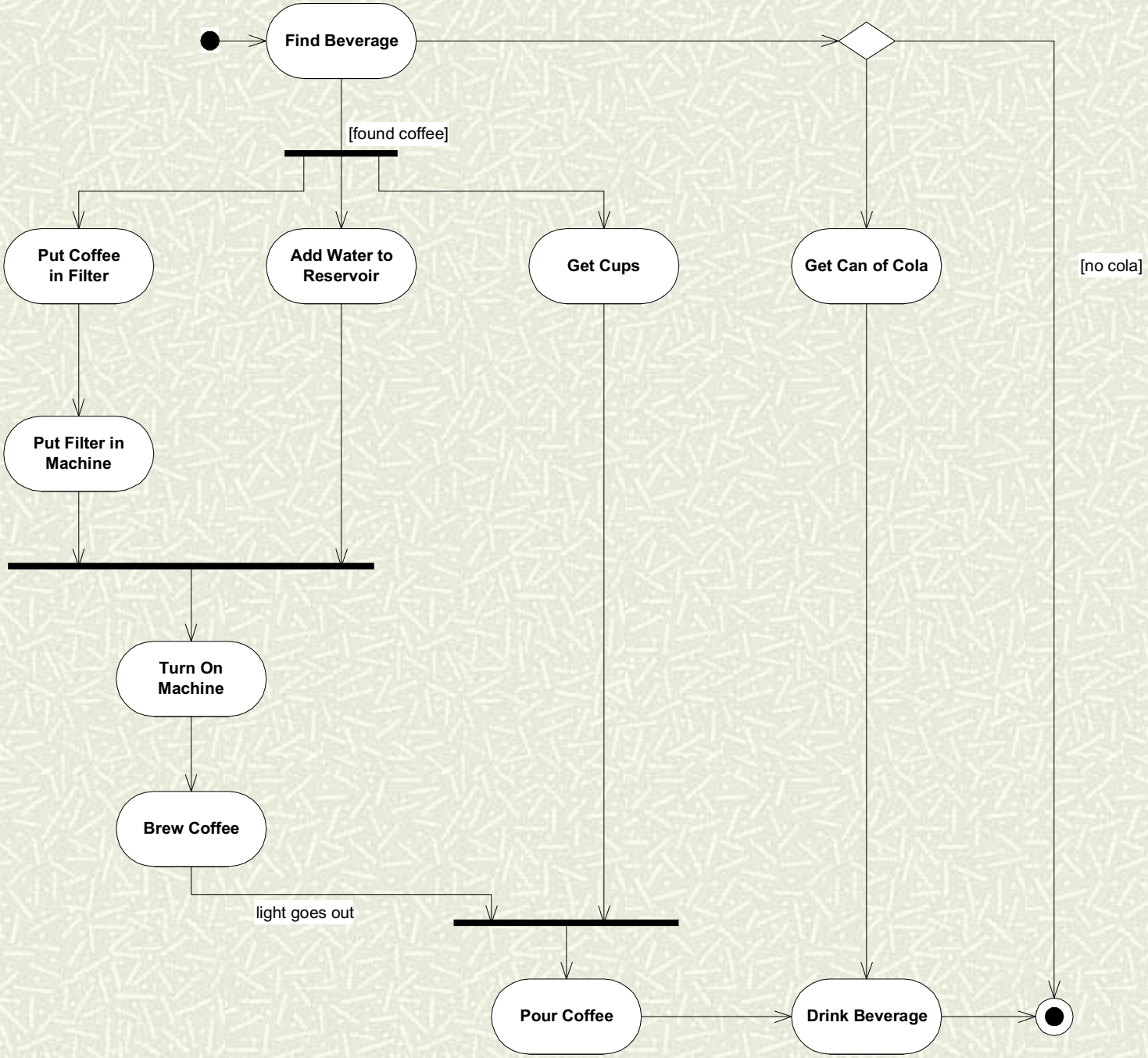


Concurrent State Diagram



Activity Diagrams

- A sequence diagram for tasks
  - instead of objects
- Show tasks and their relationships


Find Beverage
[found coffee]
Put Coffee in Filter
Add Water to Reservoir
Get Cups
Get Can of Cola
[no cola]
Put Filter in Machine
Turn On Machine
Brew Coffee
light goes out
Pour Coffee
Drink Beverage

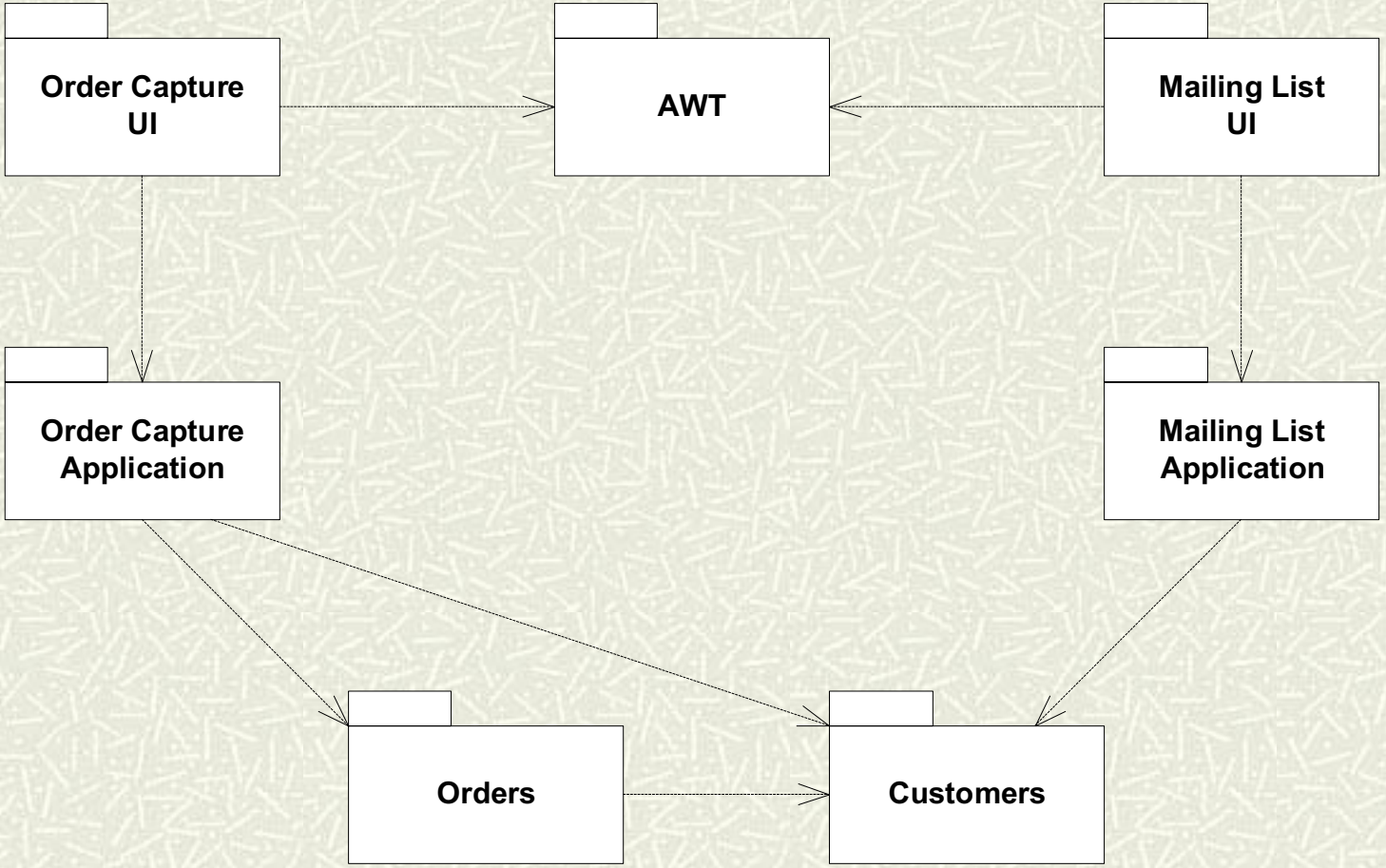
Grouping Diagrams

- Package
- Deployment

Package Diagrams

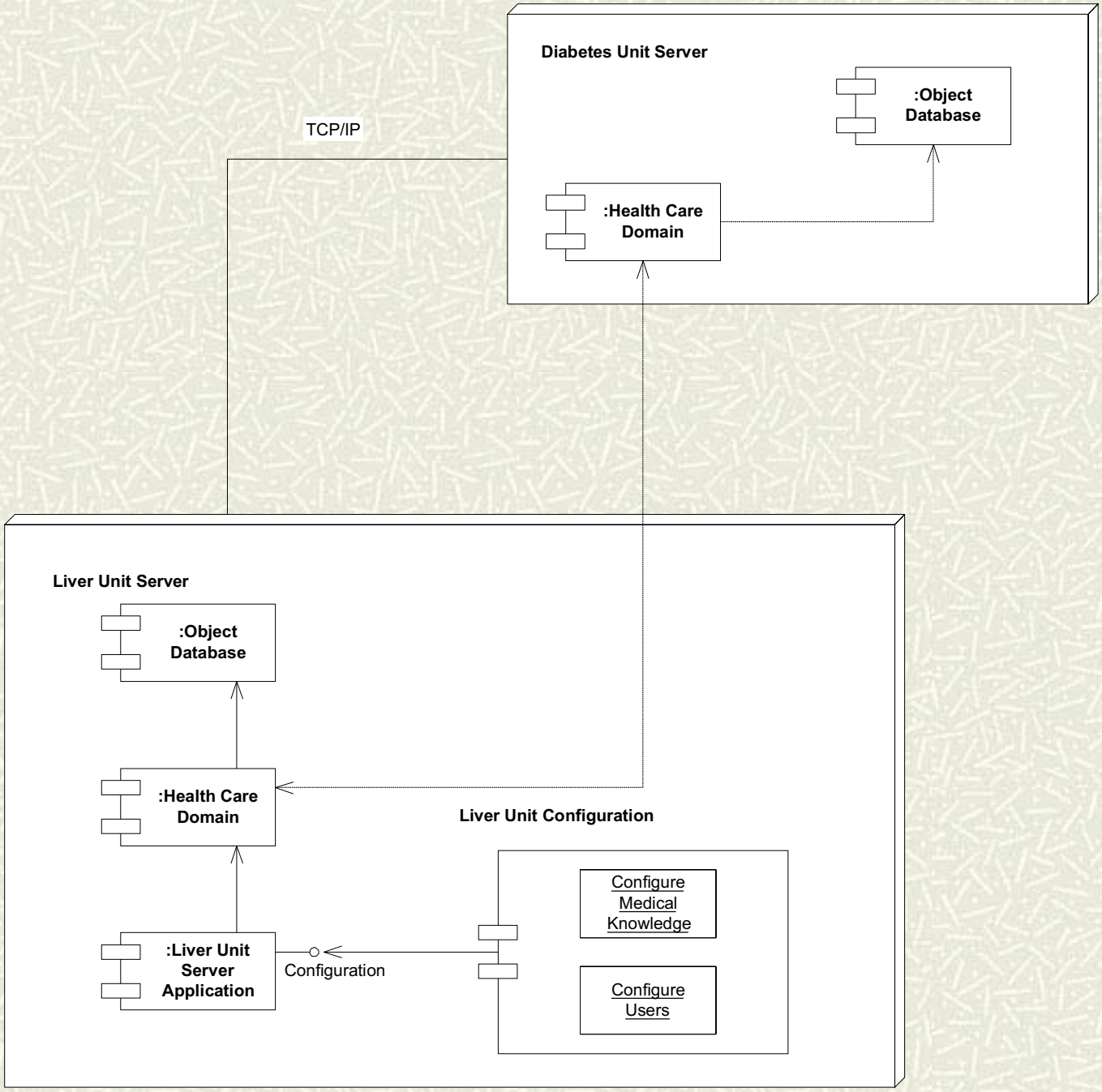
- A System Decomposition Diagram
- For Partitioning into Modules
  - often groups of classes
- Useful to show dependencies among modules

Package Diagram



Deployment Diagrams

- Depict the Relationships between components of a system


Diabetes Unit Server
:Object Database
TCP/IP
:Health Care Domain
Liver Unit Server
:Object Database
:Health Care Domain
Liver Unit Configuration
Configure Medical Knowledge
:Liver Unit Server Application
Configuration
Configure Users

Relationships

- Dependencies
- Associations
  - aggregation
- Generalization (Inheritance)

Dependencies

- Merely shows that if one element changes, then the dependent element is affected
- Denoted by --------→

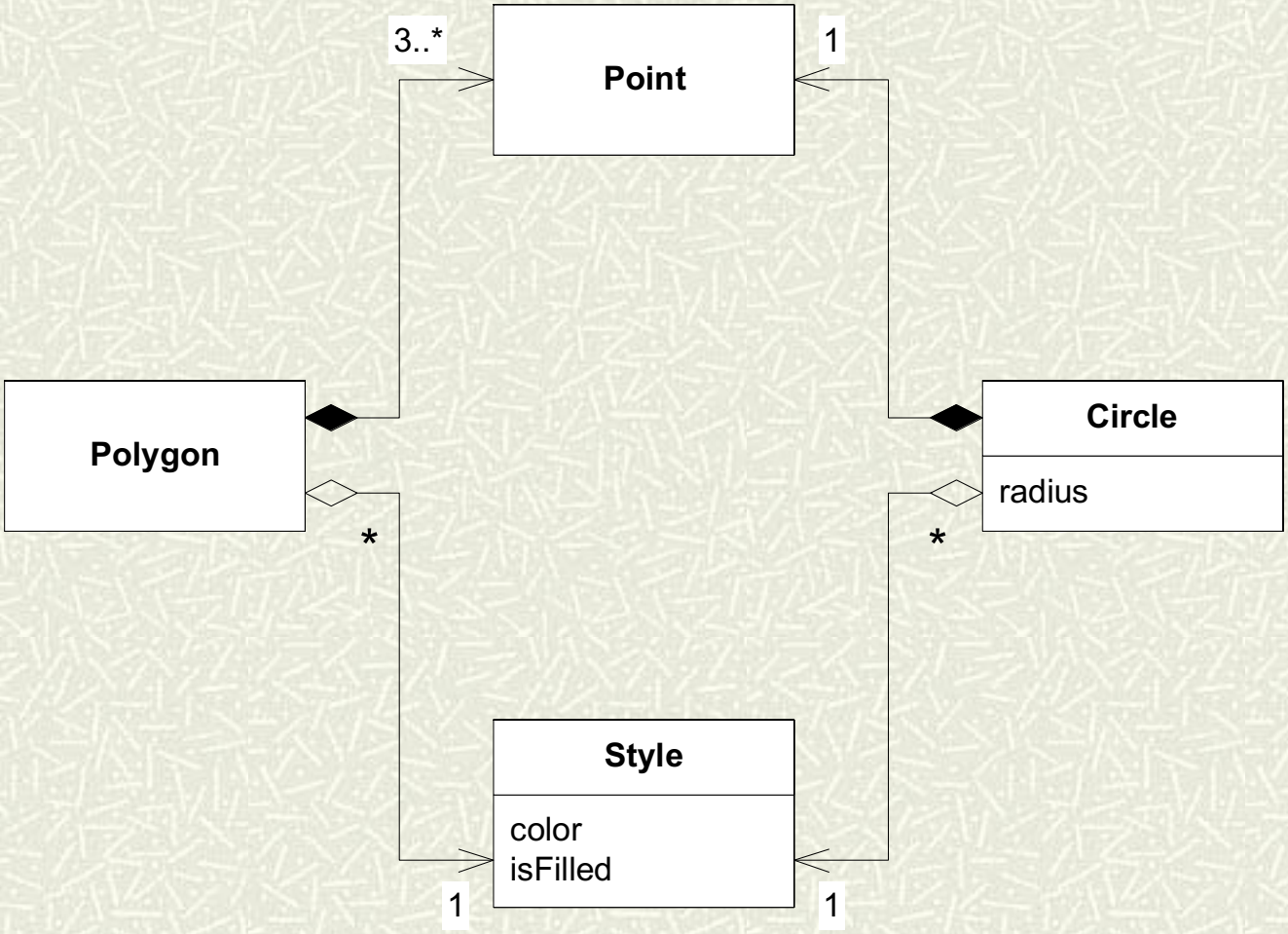
Associations

- At the most general level, indicates nothing more than some “connection”
  - unadorned straight line between two elements
  - can include cardinality

Aggregation

- A part-whole relationship
- Also called “containment” or “composition”
- Uses a diamond icon on the association line
- Composition is special:
  - means contained object is not shared
  - uses a filled-in diamond

Aggregation & Composition



Notes

- Just a Note!

This is a note. it can be placed anywhere.

Summary

- Everybody's using it
- Use what applies to your system
- Free Visio Templates available on Web
- Book: UML Distilled, Martin Fowler, AW, 1997.