

# Architectural Documentation

- View: summary data presentation
  - Graphical, tabular, textual
- Kruchten's 4+1 Views
  - Philippe B. Kruchten. "The 4+1 View Model of Architecture." *IEEE Software*, 12(6):42-50, November 1995
  - Logical, developmental, process, physical, use case
- Features, non-functional requirements, bug reporting, context, utility

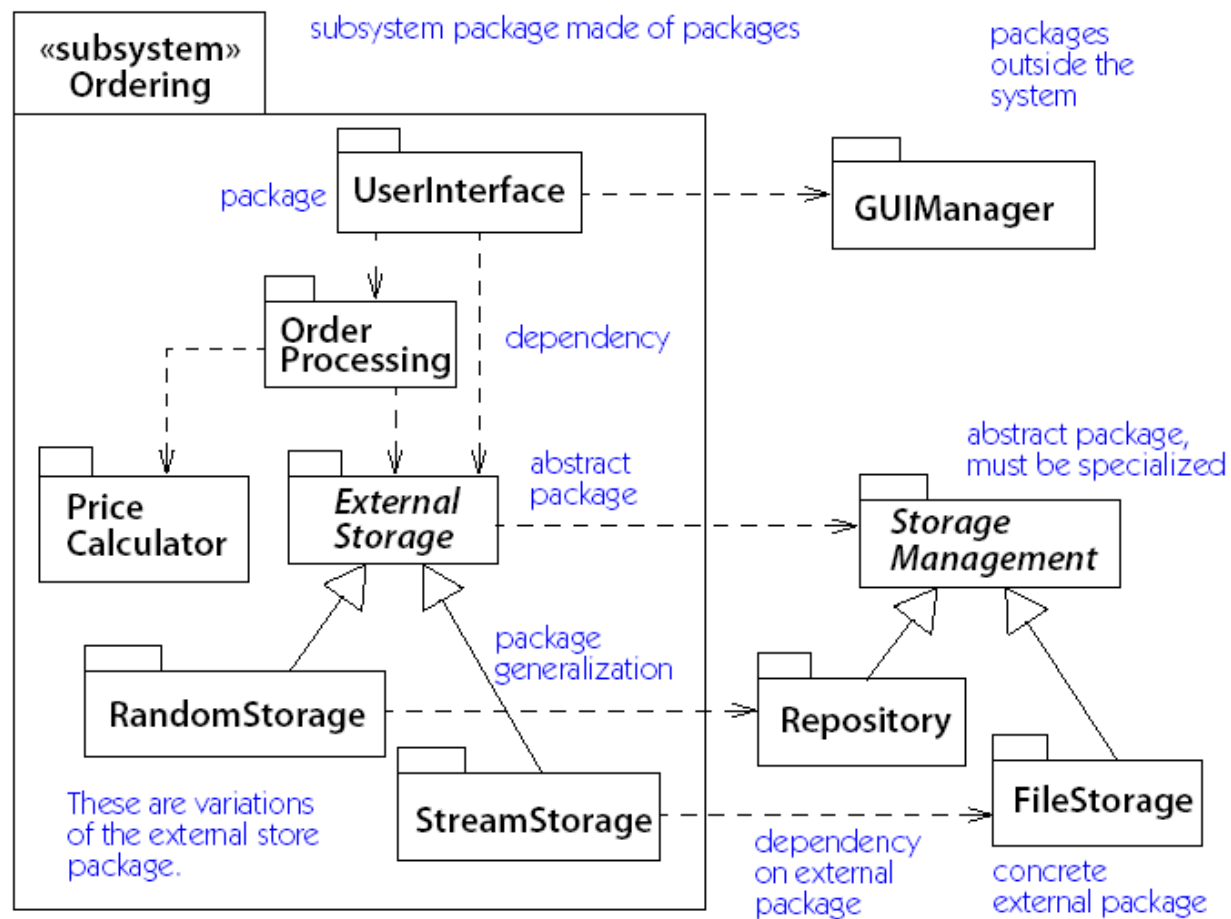
# Logical View

- Structural breakdown of computational, communicational and behavioral responsibilities
- Diagrams:
  - Box and arrow
  - Components and connectors (ADLs)
  - UML Class model diagram
  - UML Interaction Overview diagram
  - UML Collaboration diagram

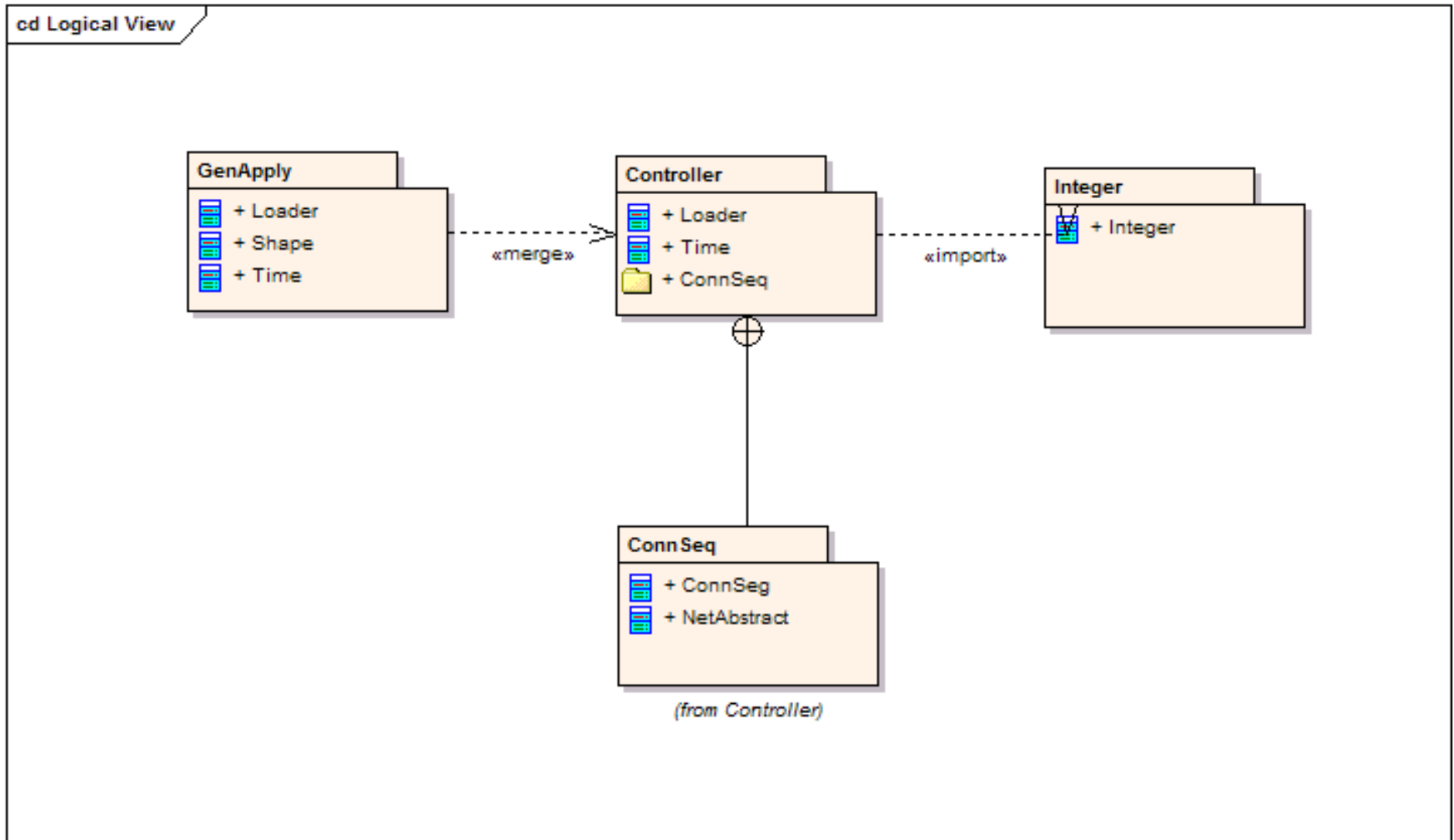
# Developmental View

- Units of source code
  - Packages, classes, subsystems, libraries, files
- Diagrams
  - UML Package diagram
  - UML Component diagram
  - CVS modules

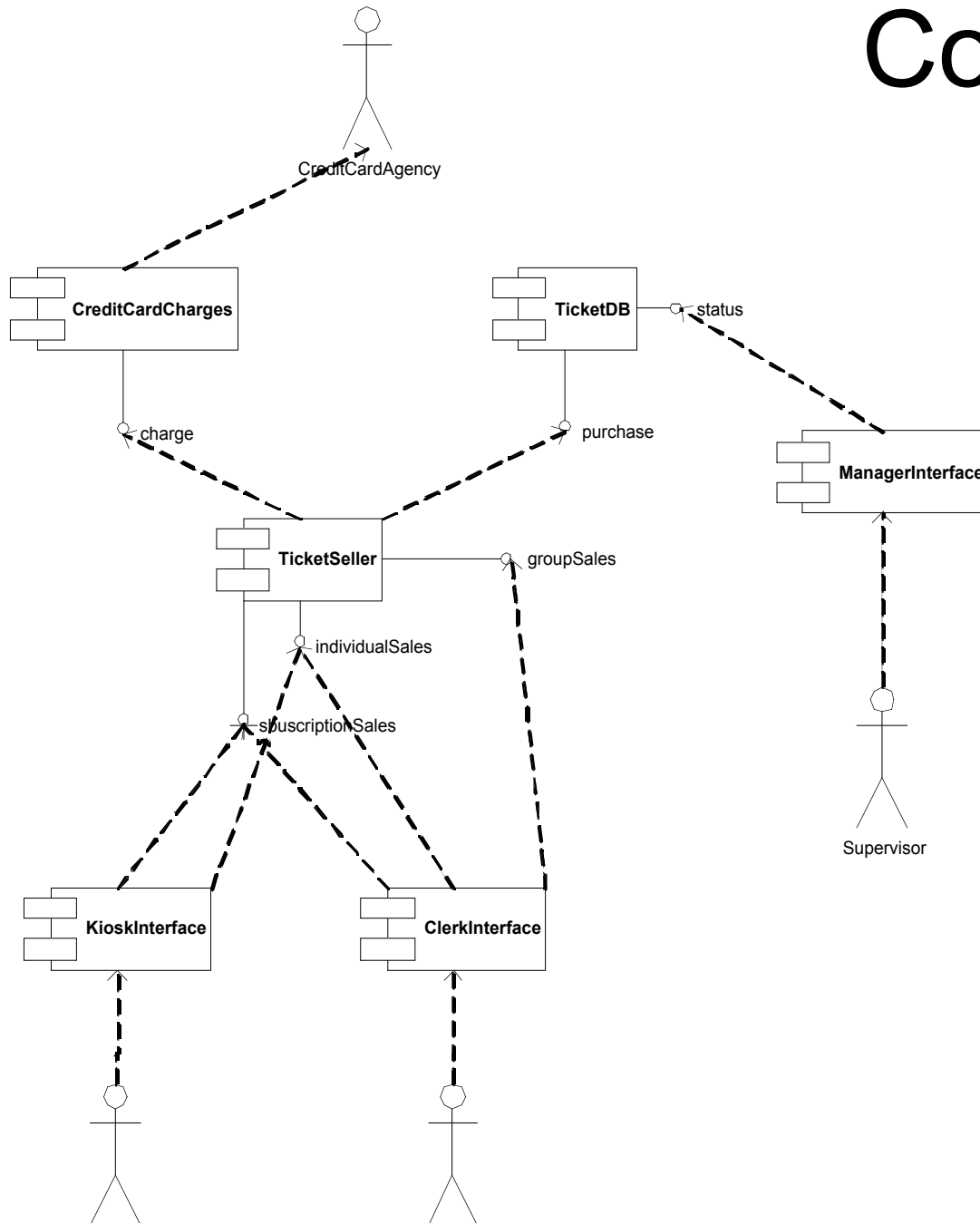
# Example Class Diagram with Packages



# Package Diagram



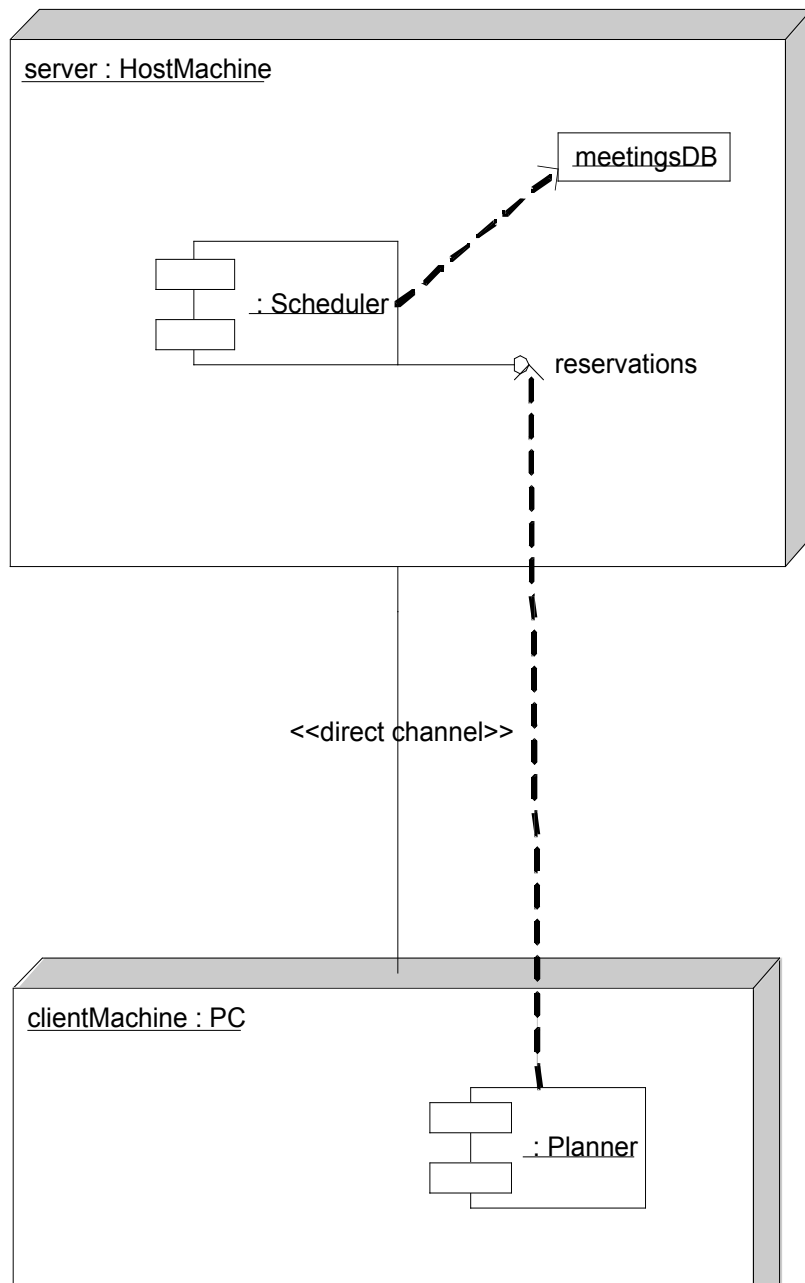
# Component Diagram



# Process View

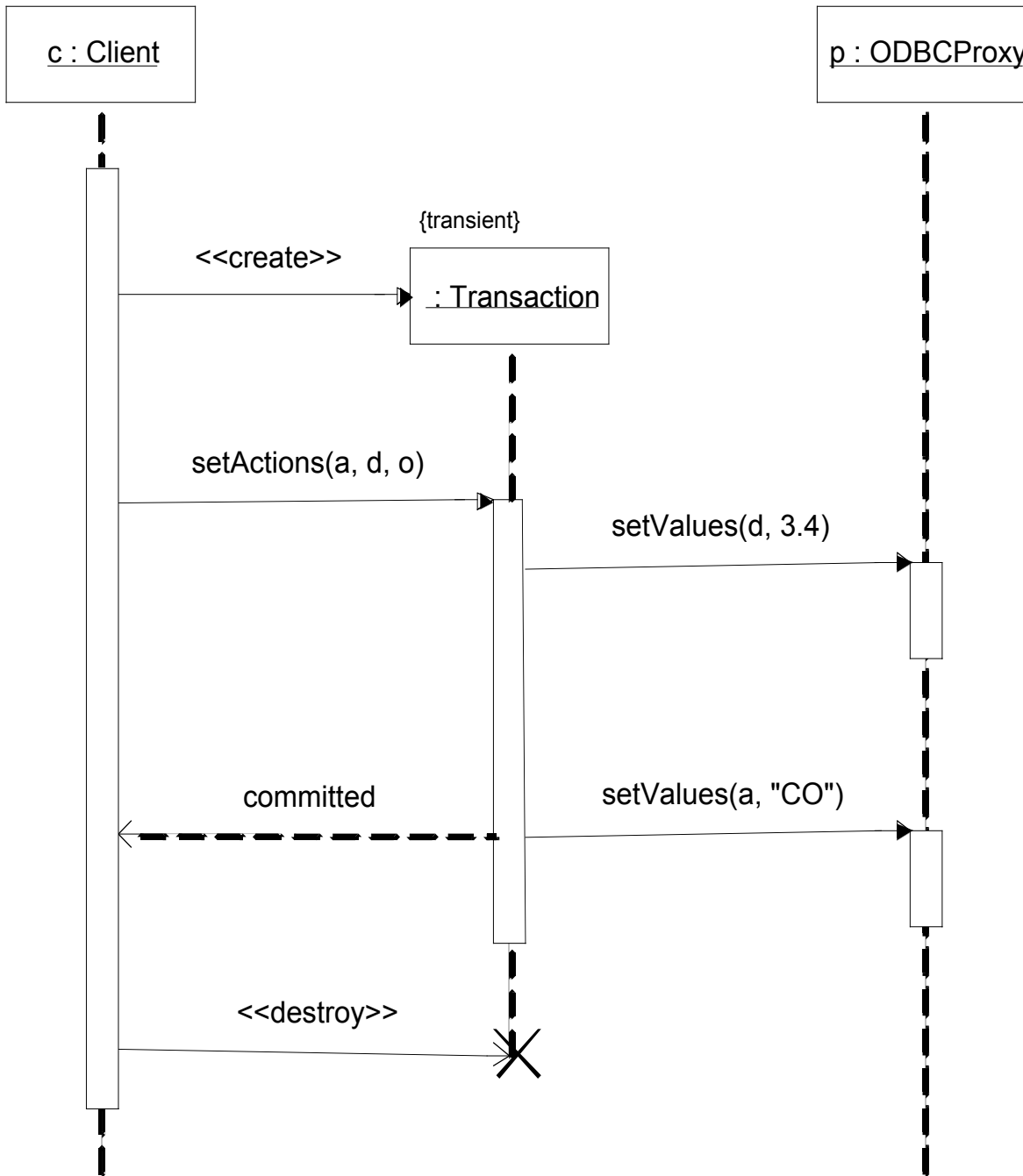
- Processes and threads into which execution is divided
- Diagrams
  - UML Deployment diagram

# Example Deployment Diagram



# Physical View

- Machines used for system execution and how processes are allocated to them
- Diagrams
  - UML Deployment diagram
  - UML Sequence diagram

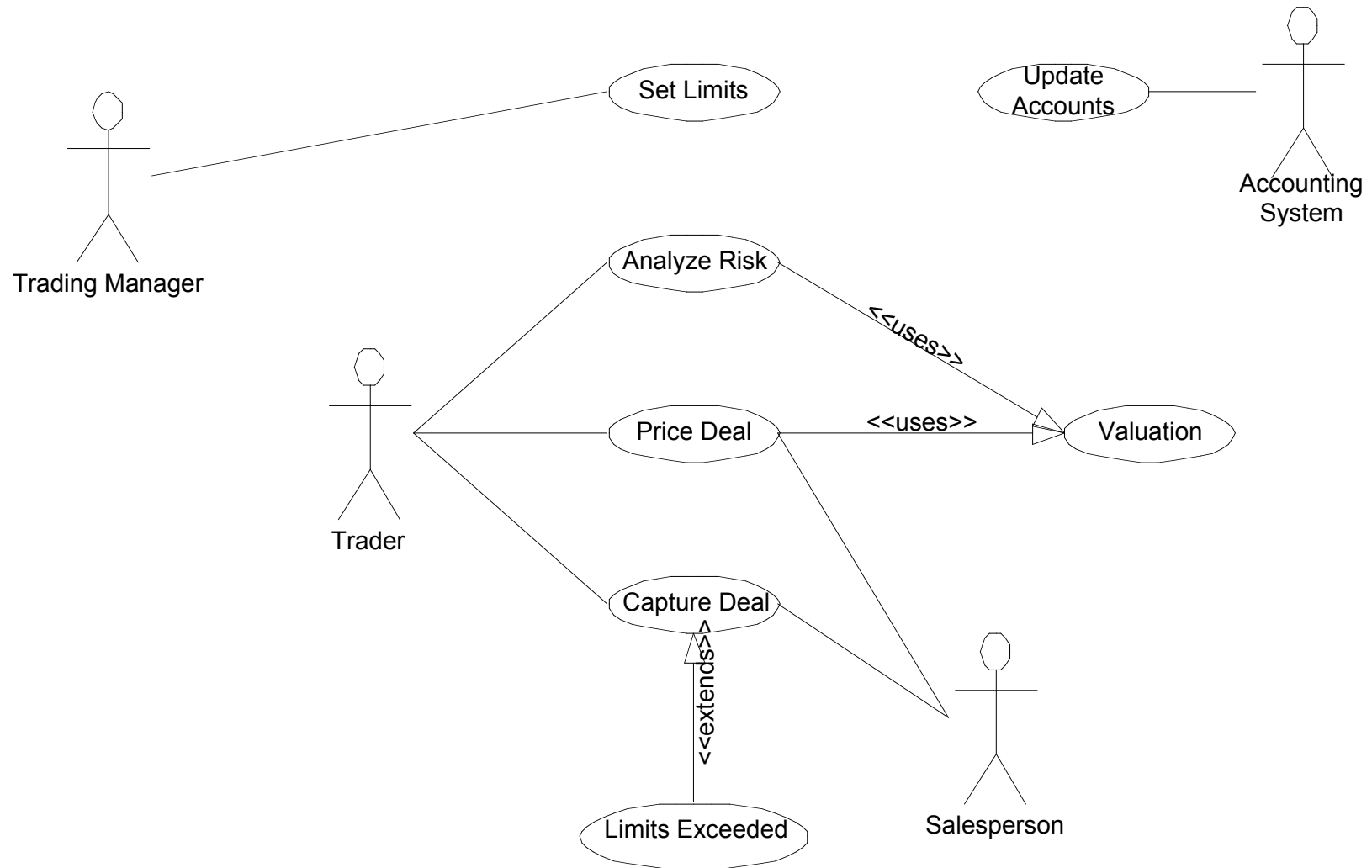


# Example Sequence Diagram

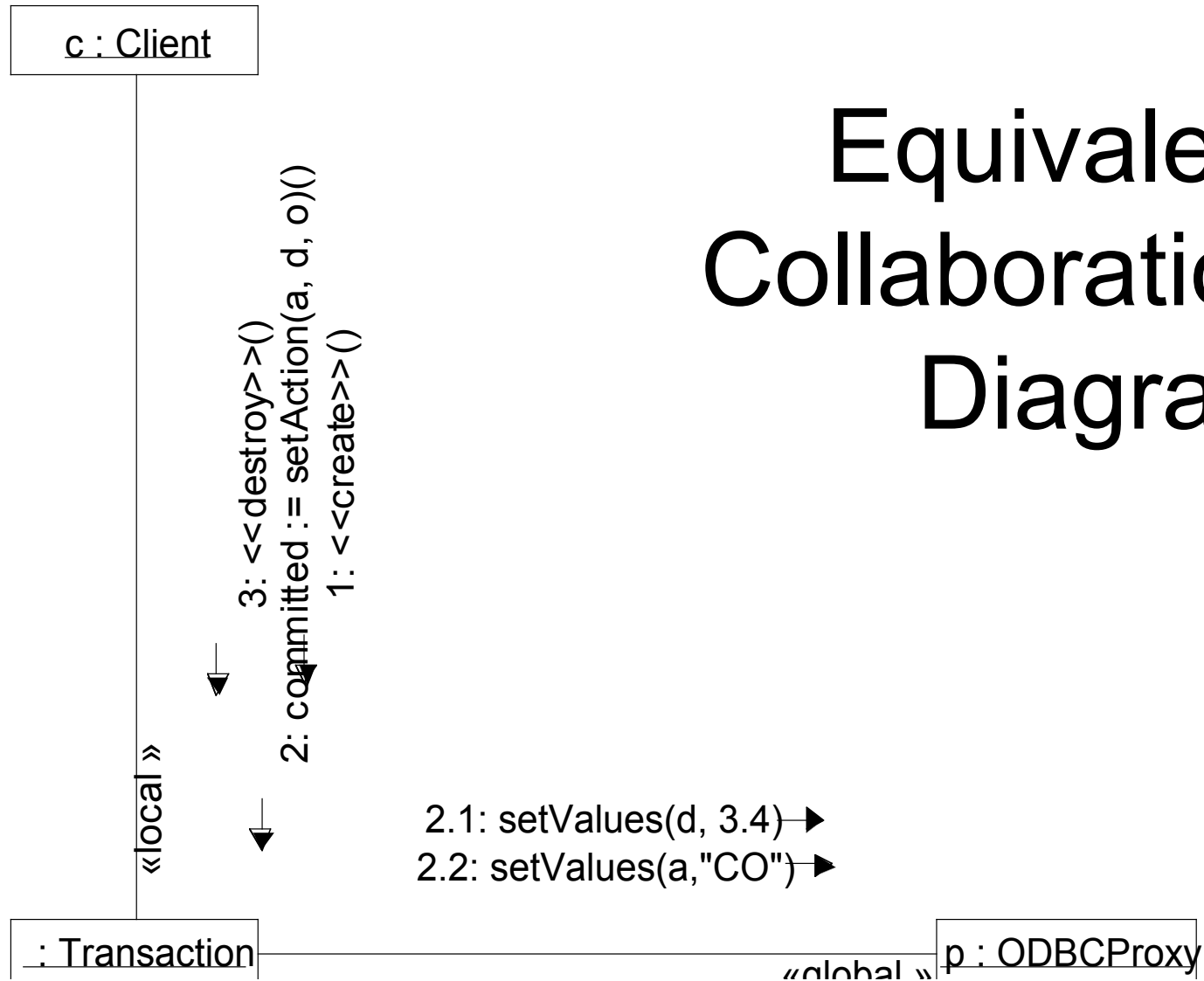
# Use Case View

- Important execution sequences from the external actors' points of view
- Diagrams
  - UML Use Case diagram
  - UML Communication (Collaboration) diagram
  - UML Sequence diagram
  - UML Activity diagram
  - Structured text

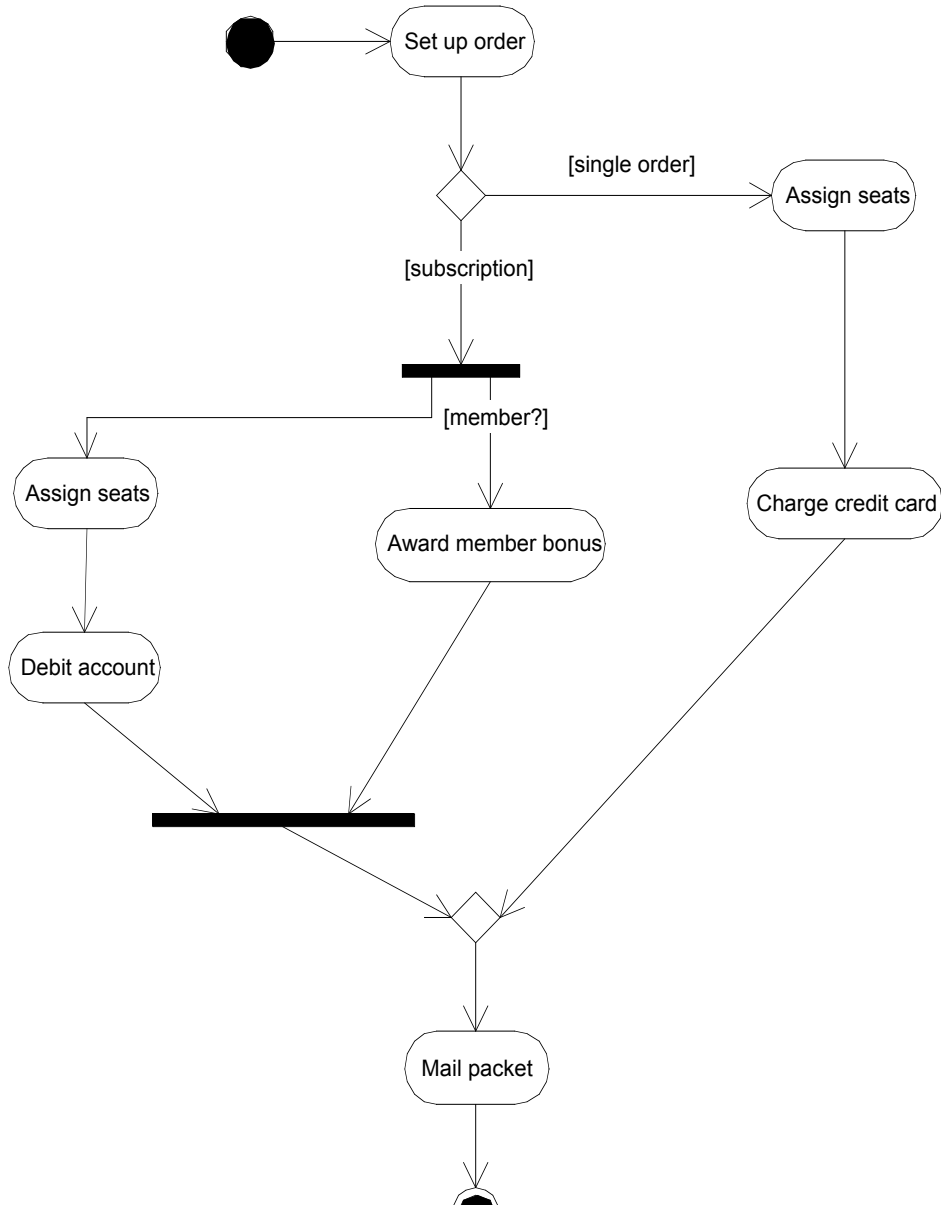
# Example Use Case Diagram



# Equivalent Collaboration Diagram



BoxOffice::ReceiveOrder

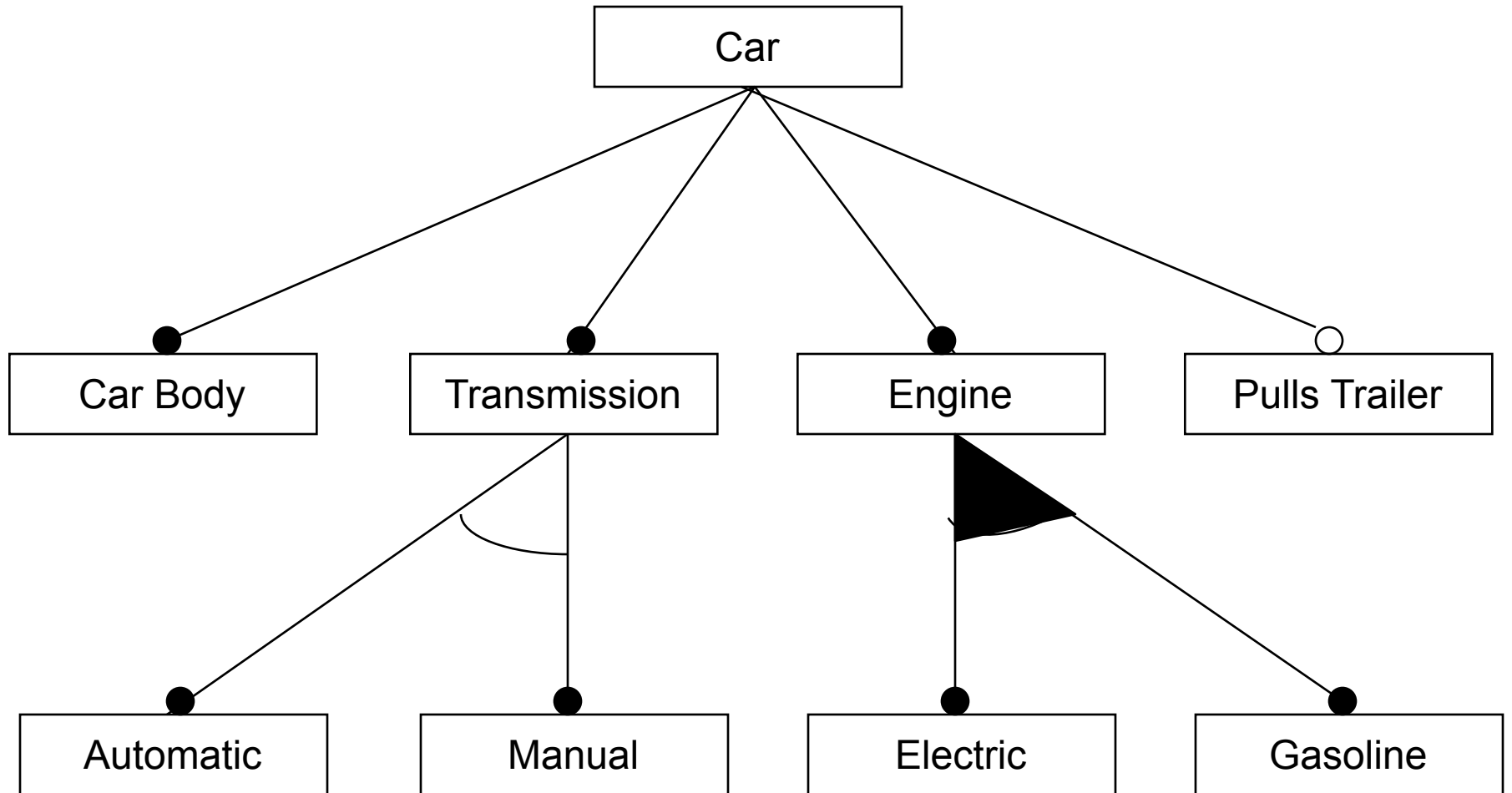


# Example Activity Diagram

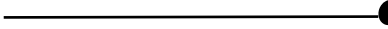
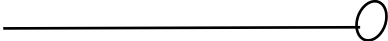
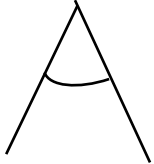
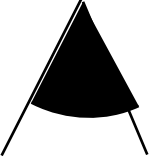
# Feature View

- Conceptual units from the user's point of view
- Diagrams
  - Feature diagram
  - Inter-feature constraints

# Example



# Notation

- Mandatory features: 
- Optional features: 
- Alternative features:
  - Choose exactly one 
- Or features:
  - Choose a subset 

# Non-Functional View

- How non-functional requirements affect the software architecture
  - Explicit tradeoffs
- Tabular text

# Bug Reporting

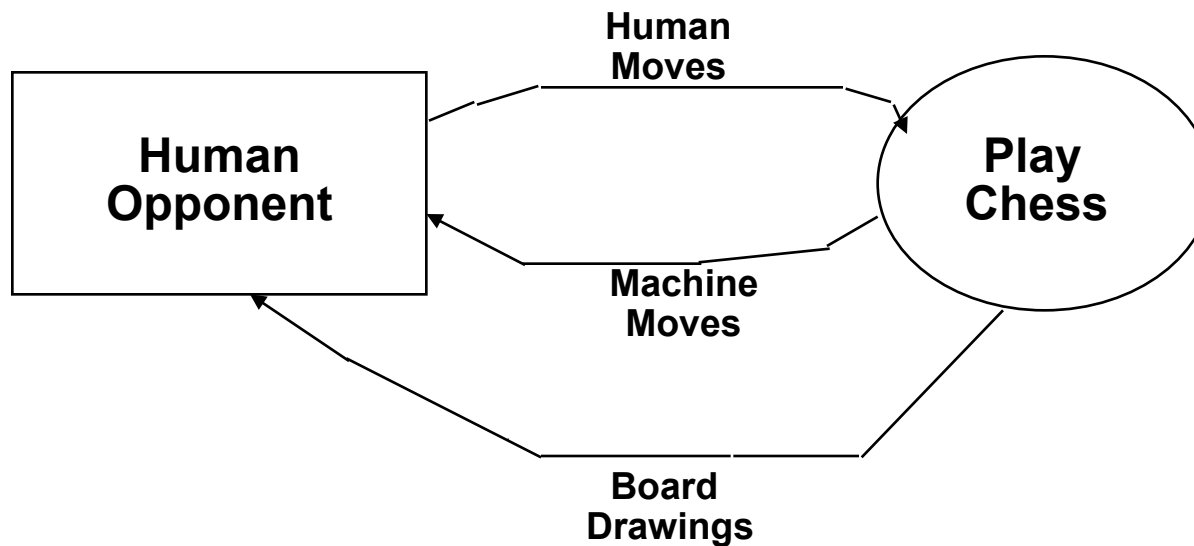
- Units to which bugs are allocated
  - Consequences with respect to Developmental view
  - Consequences with respect to Feature view
- Bugzilla components

# Context View

- Relationship of a system to its environment
  - External actors, events and percepts
- (Data flow) Context Diagram

# Context Diagrams

- The top level data flow diagram is called the context diagram
- It contains exactly one process node denoting the overall function of the system



# Utility View

- Supporting software
  - Installation scripts
  - Log file analysis
  - Statistical processing
- Tabular text

# Conclusion

- No such tangible thing as THE architecture
- Instead there are many possible views
- You should select the views to use depending on the audience and the problem you are trying to address