

# OO Methodology Overview

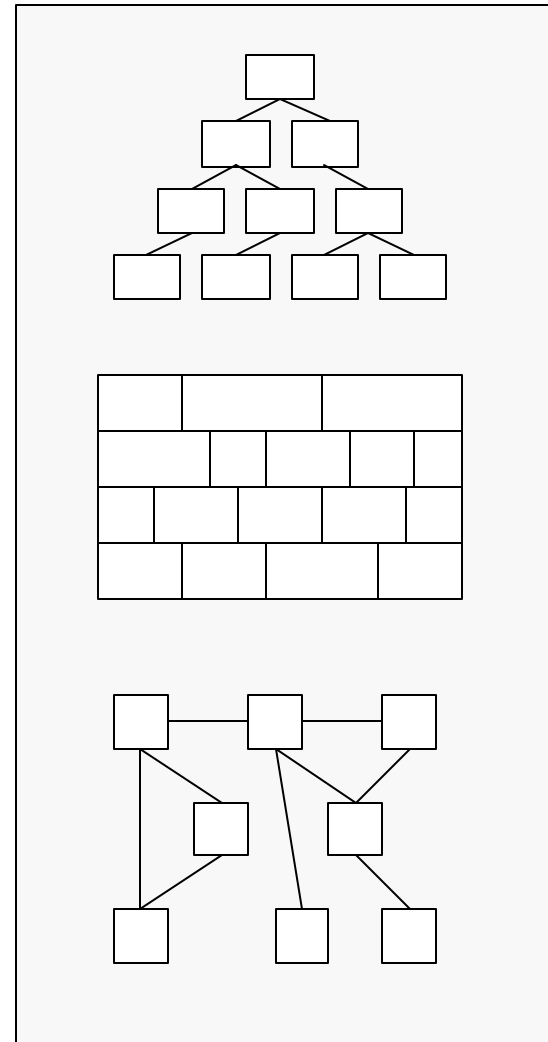
- Modeling.
  - Architectural Models.
  - Static Models.
  - Dynamic Models.
- Preview of OO Methods.
  - Requirements models.
  - Analysis models.
  - Design Models.
  - Implementation concerns.

# Modeling

- Architectural models -- Show the modular structure.
- Static models -- Model aspects that don't change.
- Dynamic models -- Indicate how things change over time.

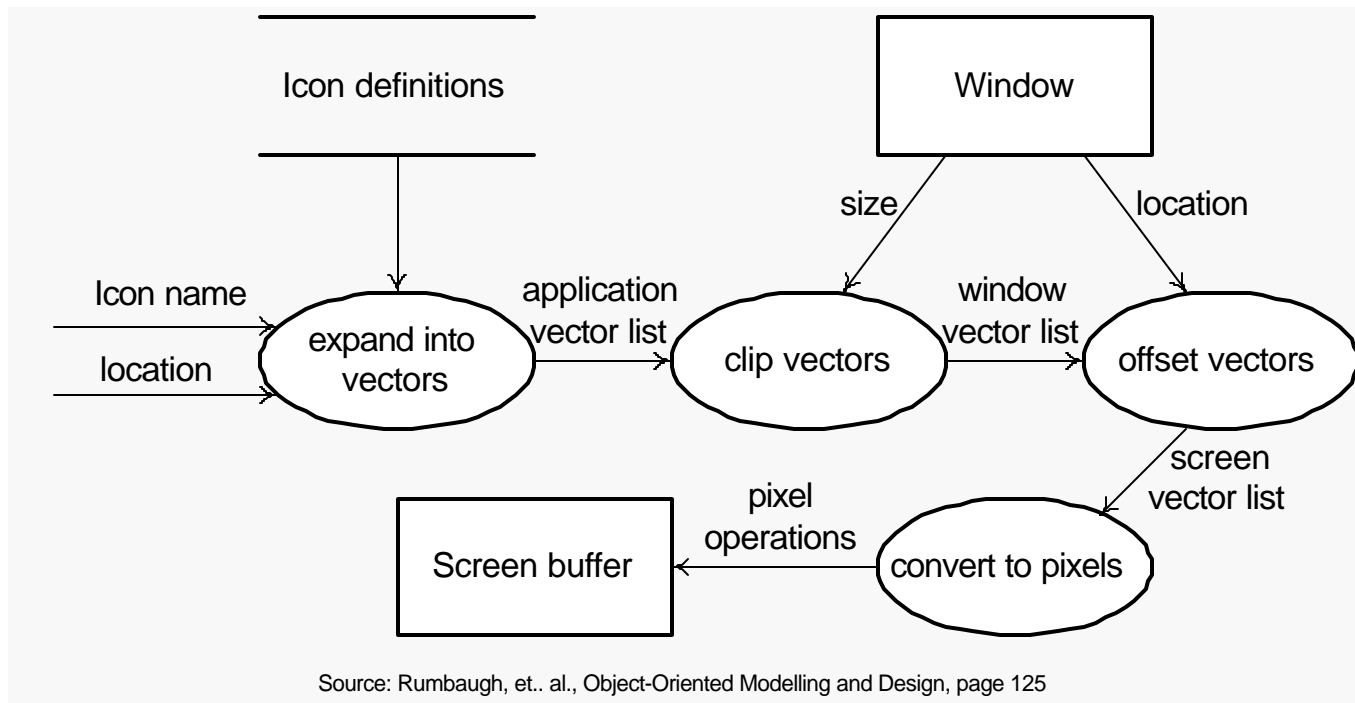
# Architectural Models

- Hierarchy (Lattice)
- Layered
- Network



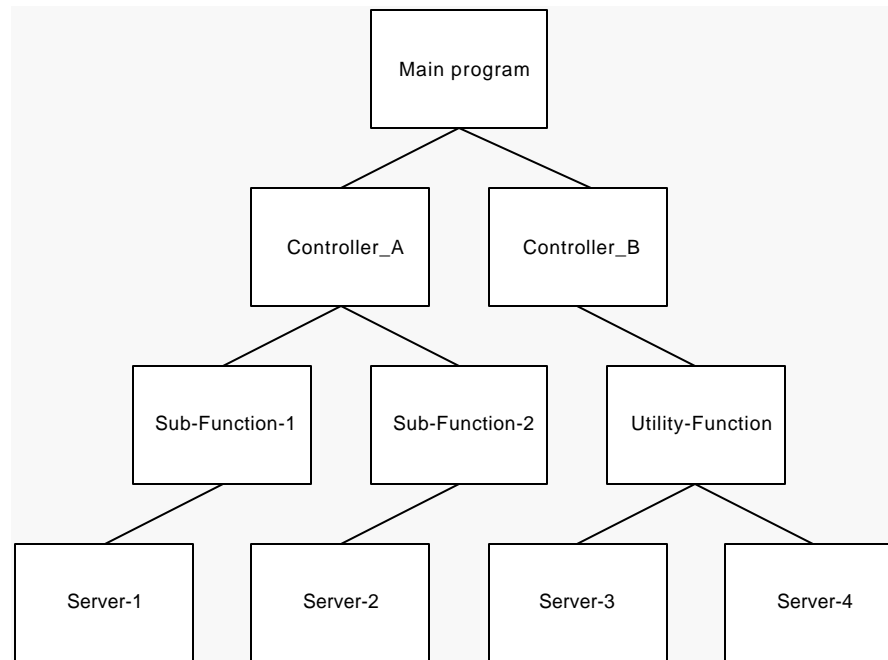
# Static Models (1 of 3)

- Data flow diagrams



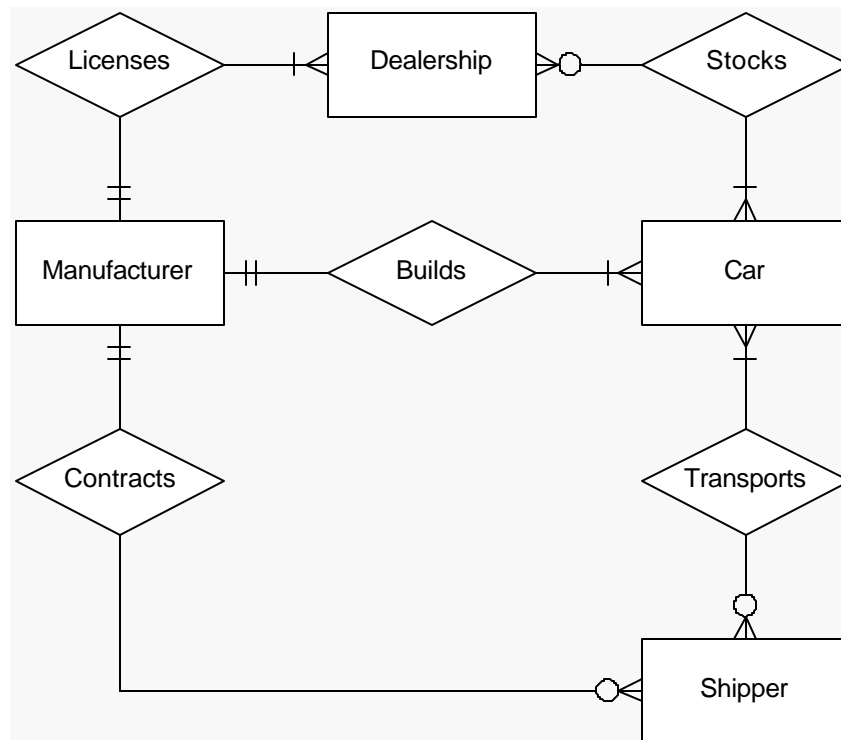
# Static Models (2 of 3)

- Structure charts



# Static Models (3 of 3)

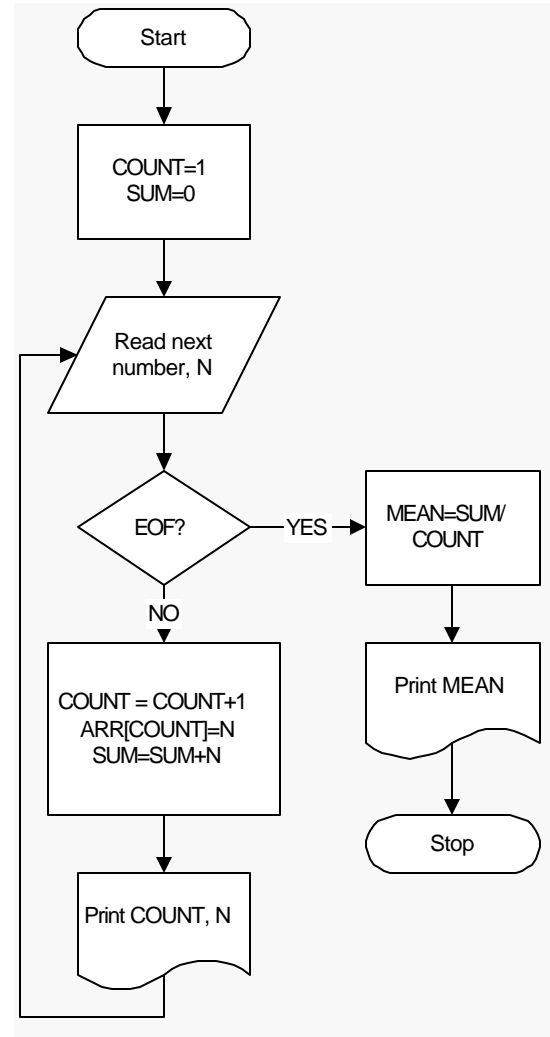
- Entity-relationship diagram



Source: Roger Pressman, Software Engineering, A Practitioner's Approach, 3rd Edition, page 26

# Dynamic Models (1 of 2)

- Flow charts



# Dynamic Models (2 of 2)

- Decision tables

Guarding the door at a fancy nightclub								
Person has had picture on cover of Rolling Stone	T	T	T	T	F	F	F	F
Person has money hanging out of pockets	T	T	F	F	T	T	F	F
Person is over 21	T	F	T	F	T	F	T	F
Get autograph	X	X	X	X				
Admit right away	X	X			X			
Call the bouncers				X				X
Make wait in line for hours			X			X	X	

# Preview of OO Methods

- Requirements models.
- Analysis models.
- Design Models.
- Implementation concerns.

# Requirements Models (Static)

- Context diagram -- Shows the boundary between the system being developed and the rest of the world.
  - Indicates which capabilities are inside the system and which ones are outside.
  - Indicates the data that flow in and out across the system boundary.
  - Indicates the other subsystems that send or receive the data.

# Requirements Models (Dynamic)

- Use cases -- Describe the ways in which the system is used by the other subsystems.
- Scenarios - Instances of a use case.

# Analysis Models (Static)

- Analysis Class Diagram -- Models the problem space
  - Classes of objects in the real world.
  - The static relationships among these classes.
    - Aggregation and Composition (part-of, containment).
    - Generalization (is-a, inheritance).
    - Association.

# Analysis Models (Dynamic)

- Interaction Diagram – Indicates a sequence of messages between objects for a particular scenario.
  - Sequence Diagram
  - Communication Diagram
- State Machine Diagram – Describes the states of an object, and the ways that these states can change over time, in response to events.

# Design Models (Static)

- Design Class Diagram – models the solution space.
  - Software classes.
  - Relationships among these classes:
    - Aggregation and Composition.
    - Inheritance.
    - Dependency.

# Design Models (Dynamic)

- Object Interaction Models – Show the message traffic among the design objects for particular scenarios.
  - Sequence Diagrams.
  - Collaboration Diagrams.
  - Activity Diagrams.
- State Chart Models – models the states and transitions of objects at the design level.

# Implementation Concerns

- Physical Architecture -- Where the objects will reside and how they will communicate with each other, and other subsystems.
- Reuse -- This is the way we will achieve higher productivity and quality goals.
- Patterns and frameworks -- This is a way to achieve reuse.
- Persistence – How to keep from losing data.