Modeling

- Help to understand the requirements thoroughly
  - Tease out questions and uncertainties
  - Holes in model reveal ambiguous and unknown behavior
- Natural language is inherently ambiguous
- “formal” language
- Clearer thinking
- Essential communications tool
- Clear aid during maintenance
- Fit on a normal page
- Consistent
  - Standards
- Decompose complex ideas
- enforcement of good structuring
- CASE tools
- end users can be taught to read/draw
- constructs that are obvious in meaning
- manipulated on a graphics screen

Decomposition Diagrams

- Organization of company, software
- Each level reveals more detail

Use-Case Diagram

- Object-Oriented Development
- Scope of system from a functional viewpoint
- Relationships between the user (actor) and system
- Depicts observable, user-initiated functionality
- Use-case -- Small more general scenario
- Actor, use-cases
  - Both are labeled
  - Interaction between
    - Not directed
- Actor
  - Role user plays
  - Interact with the use-case either initiating or participating
  - People, departments, external systems
- Use-Case
  - Externally required functionality
  - Accompanied by a scenario
- Draw
  - Method 1
• Identify the external events
• Relate the events to actors which participate in them
- Method 2
  • Identify the actors
  • For each actor identify the activities they are involved in

**Context Diagram**
- Structured Systems Analysis and Design
- Diagram 0
- System scope from information perspective
- Relationships between the external entities and system
- External entity
  • Producer or consumer of information
- Flow of information from source to sink through directed arcs
- System is a single process
- External entity
  • Box – labeled
- Process – labeled circle
  • Operates on or transforms data
- Flow – directed arc
- Data store
  • Labeled parallel lines
- Rule
  • Entities can not communicate directly
    • Irrelevant to the system