



# 7. Complex Models

---

## Overview

- 7.1 Issues
- 7.2 Object Modeling Technique
- 7.3 Use Case Approach



## 7.1 Issues

---

- Complex systems require methods which provide a rich repertoire of concepts and tools
- Effective application of such techniques demands
  - specialized training
  - deep understanding of fundamentals
  - project by project tailoring
  - gradual acquisition of expertise
- Multiple models must be integrated formally
- Graphical representations are not always economical and intuitive
- Established methods do not always deliver



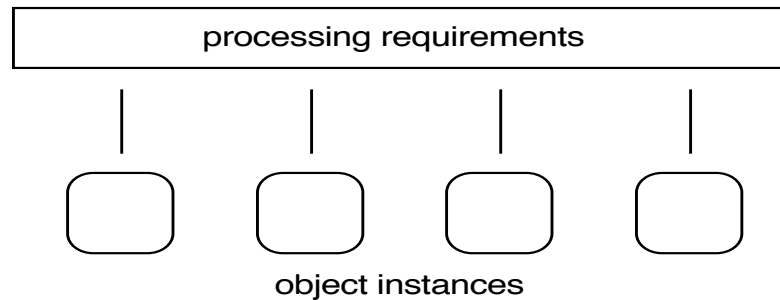
## 7.2 Object Modeling Technique

---

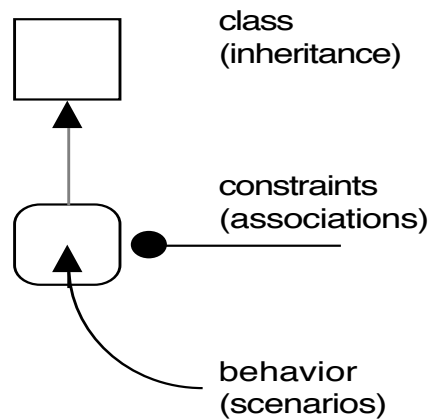
- *Object Modeling Technique (OMT)* (Rumbaugh et al) attempts to integrate several traditional methods with object-oriented analysis
- Object model
  - application-specific information (structure) is captured in terms of objects having attributes and operations
  - classes and inheritance are used to generalize objects
  - links and associations define logical relationships among objects and classes
- Dynamic model
  - the dynamic behavior of objects is captured using state machine models
- Functional model
  - the processing of environmental inputs is captured by means of a dataflow model

# Model Overview

## Functionality



## Concepts

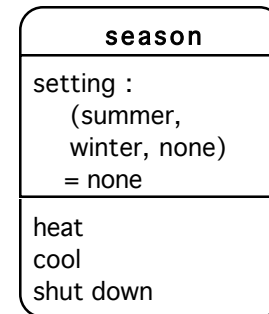




# Objects

---

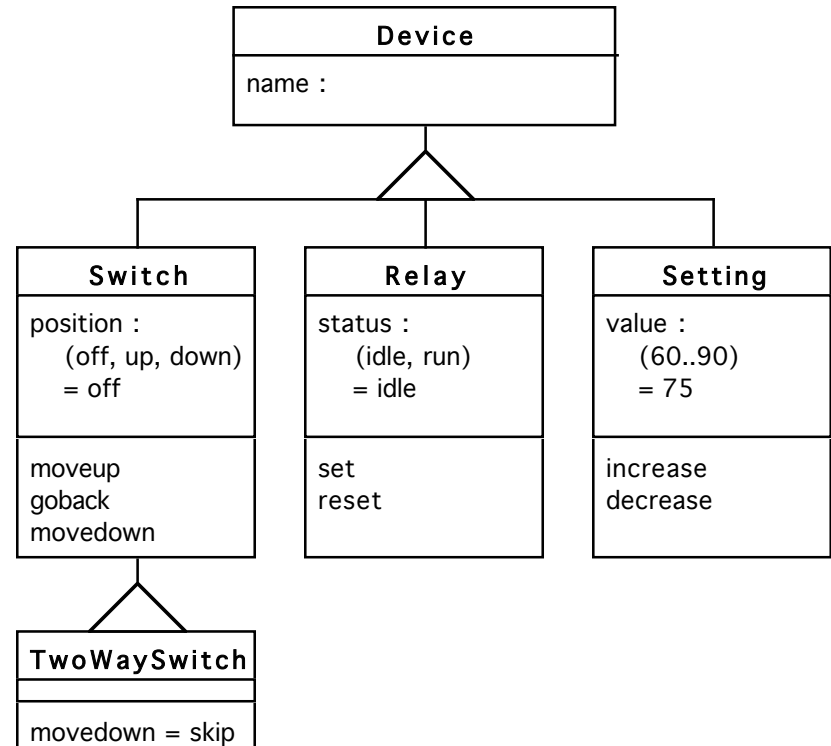
- An object is characterized by its visible attributes and the operations (services) it provides
- Objects often are immediately identifiable in the application domain
- Initial definitions may be misleading and may need rethinking



- What does it do?
- What happens in Australia?

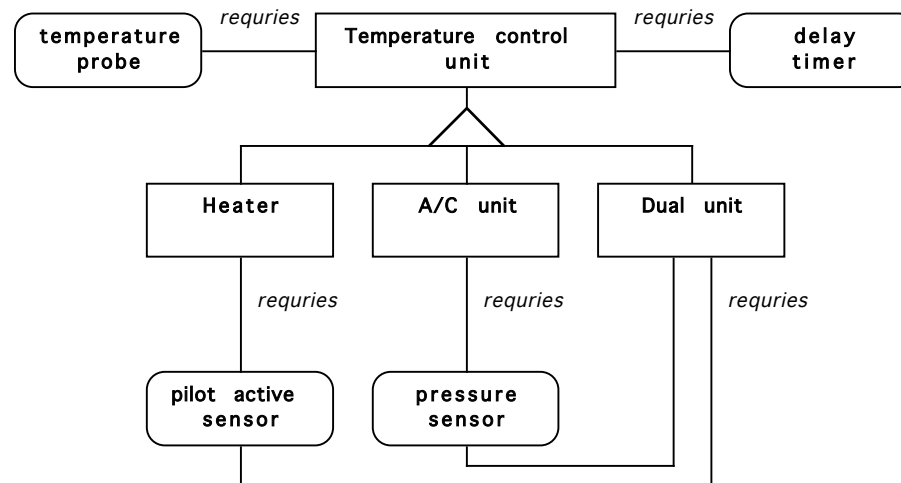
# Classes and Inheritance

- The class provide the means for object generalization, from instance to concept



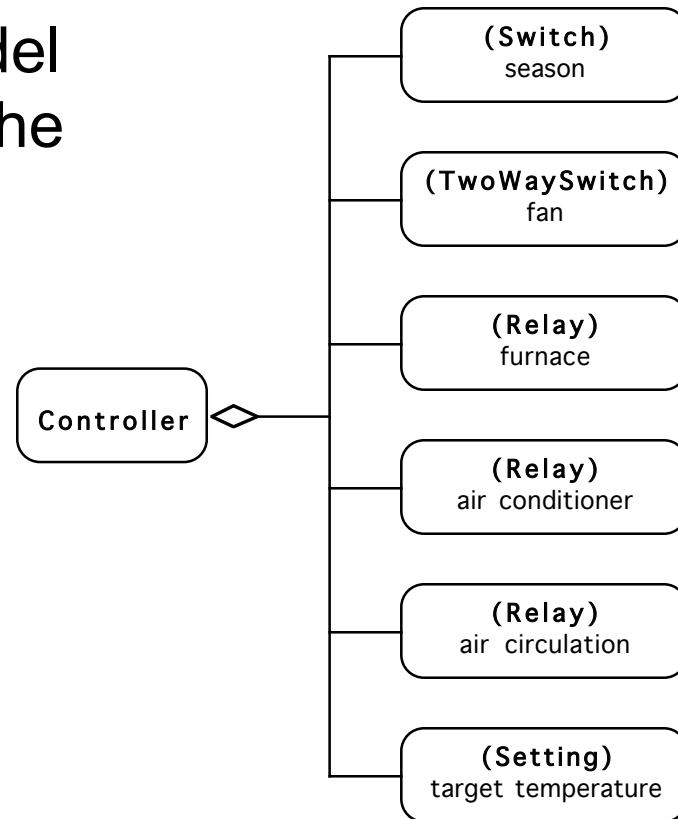
# Constraints and Associations

- Established data modeling techniques provide the means for defining semantic constraints existing in the application
- Such models help analysis



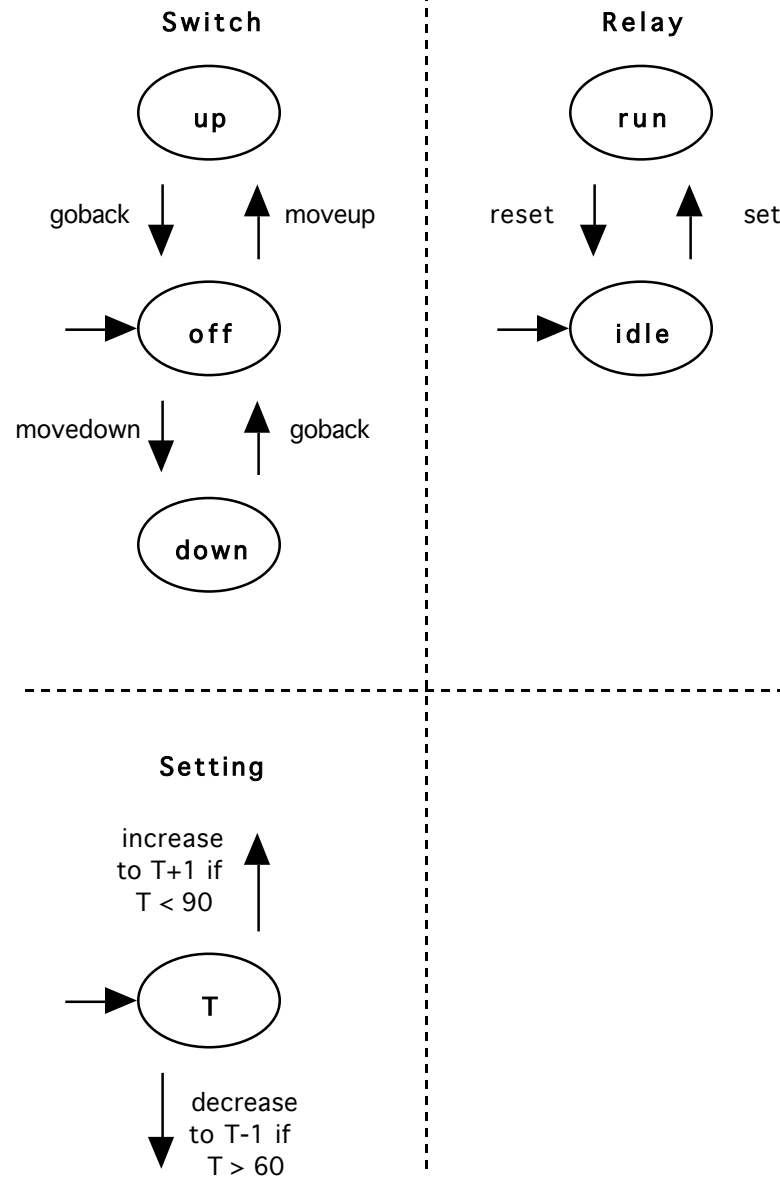
# Structure

- Object instances model the abstract state of the system

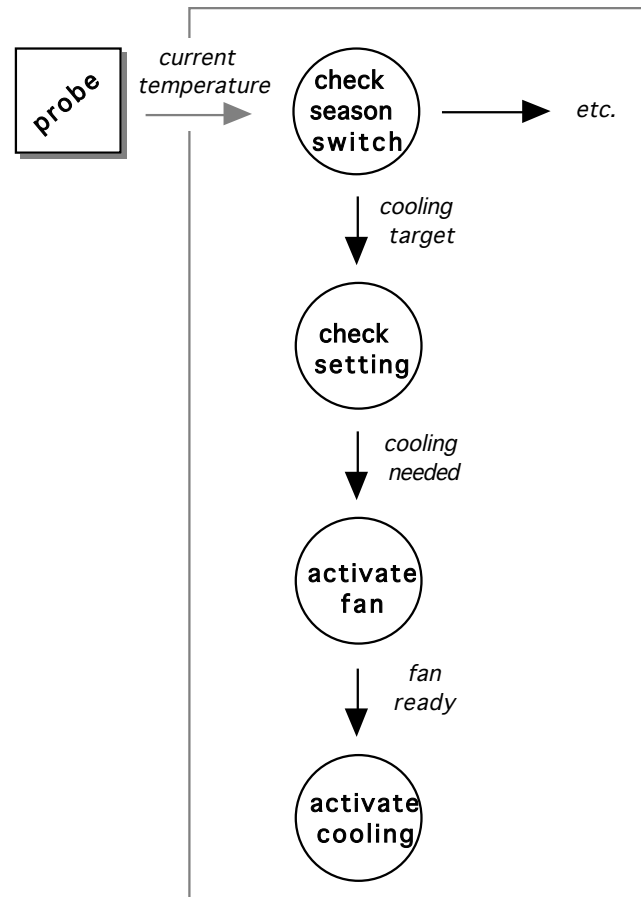




# Dynamic Model



# Functional Model





# Strengths of OMT

---

- Comprehensive application analysis
- Powerful object-oriented model
- Inclusion of relational concepts (semantic constraints)
- Reliance on established models



# Concerns with OMT

---

- Complex graphical notation
- Lack of precise formal definition
- Weak integration among models
- Inadequate treatment of the environment
- Use of models whose effectiveness is questionable (dataflow)
- Unrealistic expectations regarding a direct transition to design



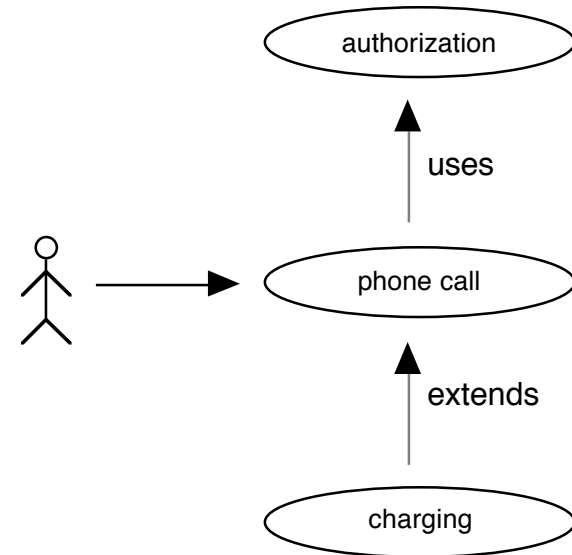
## 7.3 Use Case Approach

---

- *Use Case Approach* (OOSE) (Jacobson et al) combines object-oriented modeling with a strong emphasis on processing scenarios
- RDD (requirements model)
  - interfaces
  - domain object model
  - use case model (scenarios)
- SRS (analysis model)
  - object-oriented model of the functionality

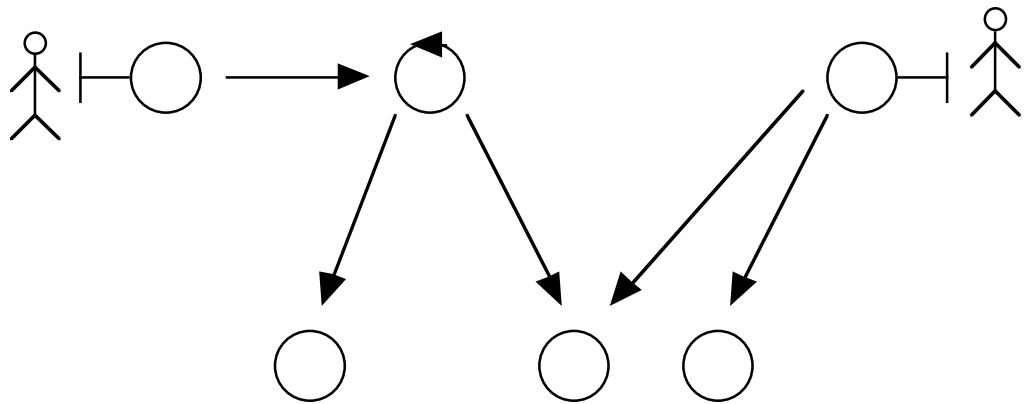
# Use Case Model

- Actors
  - model the environment and the users
  - initiate activity by providing stimuli
  - can be primary or secondary
- Use cases
  - are complete courses of action initiated by actors (basic or alternative)
  - can be extended by (interrupt analogy) or use other use cases (call analogy)



# Analysis Model

- Objects are divided into three categories
  - interface
  - entity
  - control





# Strengths of OOSE

---

- Comprehensive application analysis
- Emphasis on processing scenarios and scenario composition
- Reliance on simple forms of established models
- Powerful object-oriented model
- Emphasis on the development process





# Concerns with OOSE

---

- Complexity and cost associated with developing the domain object model
- Optimistic expectations regarding the transition to design