

CS 485  
Advanced Object Oriented Design  
Design  
Spring 2019

# Objects

- Old view
  - "find the nouns and verbs" - *many old OO design books*
  - still useful starting point
  - limited view
- Data + responsibilities
  - give your objects jobs to do

# Example, p11, Shalloway

- Students moving around a conference
  - structured programming\*
    - The driver carefully directs each person to the next spot
  - OO Design
    - ask what you want, not tell how to do it.
      - helps to insulate you from change.

# Example, cont.

- OO Design

# Design Perspectives

- Conceptual
- Specification
- Implementation



# Exercise

- Find the classes! Find the responsibilities!
- CRC Cards
  - Class-responsibility-collaboration

Class Name	
Responsibilities	Collaborators

<http://agilemodeling.com/artifacts/crcModel.htm>

<i>Order</i>	
<i>Check items are in stock</i>	<i>Order Line</i>
<i>Determine the price</i>	<i>Order Line</i>
<i>Check for valid payment</i>	<i>Customer</i>
<i>Dispatch to delivery address</i>	

<http://www.cs.unc.edu/~stotts/145/CRC/crc.html>

<http://c2.com/doc/crc/draw.html>

[c2.com/doc/oopsla89/paper.html](http://c2.com/doc/oopsla89/paper.html)  
Kent Beck & Ward Cunningham

<http://userpages.umbc.edu/~cseaman/ifsm636/lect1108.pdf>

# Process

- Brainstorm
  - find all the nouns and verbs
- Identify classes
  - you'll throw a bunch away and add some back
- Role play
  - run through various scenarios for the software
  - use cases



# Example - Shalloway, students at a conference

## Traveler

Responsibilities	Collaborators
Know the current location	Instructor
Know next classroom	ClassroomMap
Move to next classroom	Path

## ClassroomMap

Responsibilities	Collaborators
Know locations of classrooms	Traveler
Find path between Classrooms	Classroom
	Path

## Class CardReader

Responsibilities	Collaborators
Tell ATM when card is inserted	ATM
Read information from card	Card
Eject card	
Retain card	

<http://www.math-cs.gordon.edu/courses/cs211/ATMExample/CRCCards.html#CardReader>

[http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE\\_517\\_Fall\\_2007/wiki2\\_5\\_kq](http://wiki.expertiza.ncsu.edu/index.php/CSC/ECE_517_Fall_2007/wiki2_5_kq)

# Exercise

- You need to build a software alarm clock!
- The alarms can be visual, audible, and vibrate alarms
- The clock must maintain many alarms (time and type(s), note)
- The clock is either 12 or 24 hour.
- Uses either metric or imperial seconds.
- Alarms are written to / read from a file.
- Alarms can be snoozed, snooze time is customizable for each alarm (default 9 min).

# Example

- Work in assigned teams
- How would you break this down into classes?
- What would each class be responsible for?
- 30 minutes for Brainstorming
- 10 minutes for two scenarios.

# Chapter 3 & 4

- Chapter 3
  - lays out the CAD problem
  - extract model from two different CAD systems
    - procedural system
    - object oriented system
  - translate model into machine operations via expert system
- Chapter 4
  - initial object oriented design
  - brief discussion of the flaws of this design

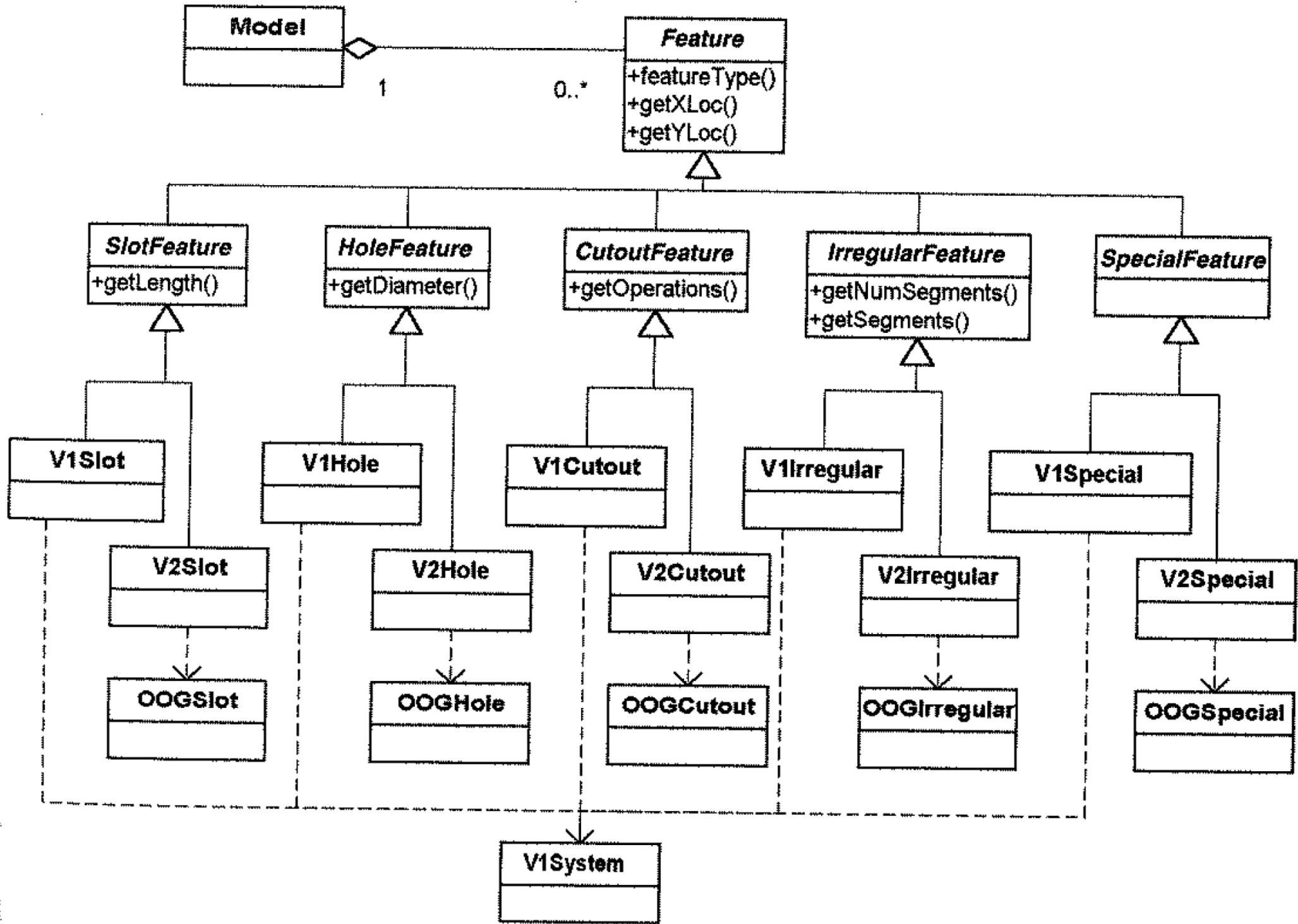


Figure 4-3 A first solution.

# Analysis - Successes

- One API for multiple backend CAD systems
- Each object has responsibilities

# Analysis - Failures

- Everything is a special case
- Redundancy among methods
- Messy / Growth from change
- Tight Coupling / Weak cohesion