

Introduction to Object-Oriented Analysis and Design (OOAD)

CSSE 374: Class 2

Shawn Bohner
Office: Moench Room F212
Phone: (812) 877-8685
Email: bohner@rose-hulman.edu



ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Q1, 2

© 2009 Shawn A. Bohner

Agenda

- ❖ **Some Design Perspectives**
 - Bridging from CSSE 371 to CSSE 374 Software Architecture and Design 1
- ❖ **Understanding the Book's Organization**
- ❖ **Introduction to Object-Oriented Analysis and Design**
- ❖ **Class Exercise**
- ❖ **Some UML Perspectives**
- ❖ **Homework Assignment**

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Why is Software Design Important?

- ❖ Size
- ❖ Complexity
- ❖ Constraints
- ❖ Performance
- ❖ Communication

Thinking at the Right Level

- ❖ **Abstraction** - hiding irrelevant details to the current design
- ❖ Process of component identification is top-down, **decomposing** the system into successively smaller, less complex components
- ❖ Process of integration, which is bottom-up, building (**composing**) the target system by combining components in useful ways



Why do you suppose that Larman does not stress learning UML?

Elaboration and Refinement...

- ❖ Starting with Abstract Requirements
- ❖ Successively Elaborate and Refine them into specifications, models, and ultimately implementation



ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Key Questions

Responsibility-driven Design

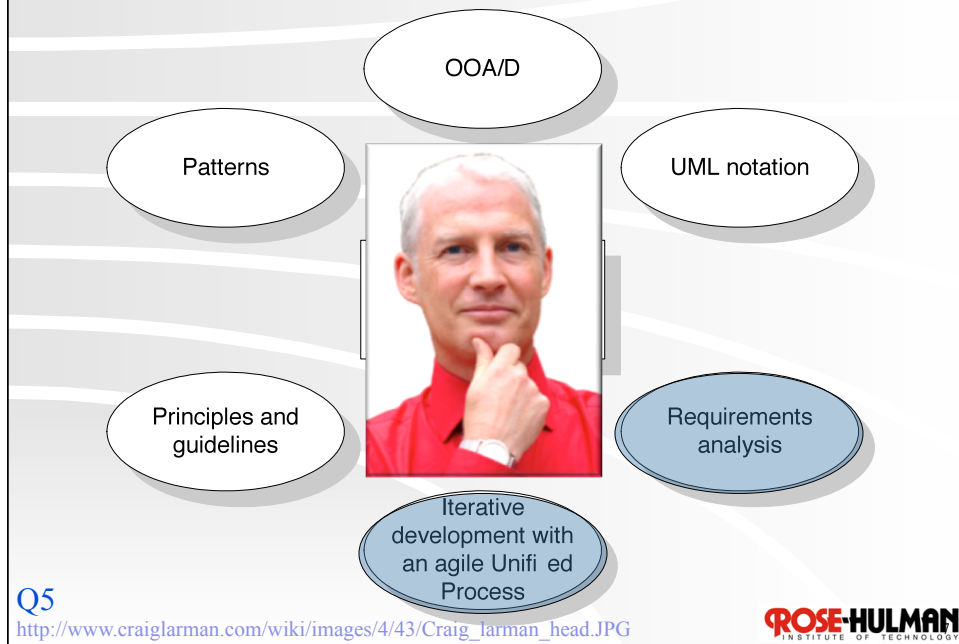
1. How should responsibilities be allocated to classes?
2. How should objects collaborate?
3. What classes should do what?

Guided by patterns

Q4

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Topics Covered in Book



Assigning Object Responsibilities

A critical ability in Object-Oriented development is to skillfully assign responsibilities to software objects.

Analysis versus Design

❖ Analysis

- Investigation of the problem and requirements, rather than a solution

❖ Design

- A conceptual solution, rather than its implementation
- Excludes low level details
 - Often represented in code



Q7

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Analysis and Design Concepts

Analysis Concept



domain concept

representation in an
object-oriented
programming language

Design Concept



```
public class Plane
{
    private String tailNumber;
    public List getFlightHistory() {...}
}
```

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Quick Example: Dice

❖ Define Use Cases

- Play a dice game: Players requests to roll the dice. System presents results: If the dice face value totals seven, player wins; otherwise player loses

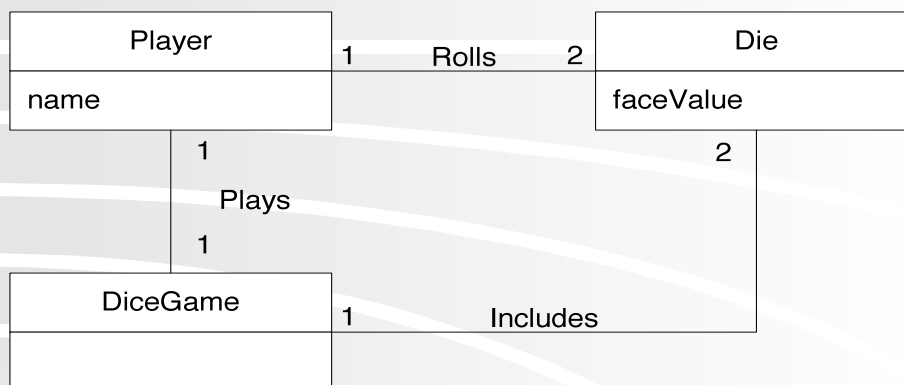
❖ Define a Domain Model

❖ Assign Object Responsibilities, Draw Interaction Diagrams

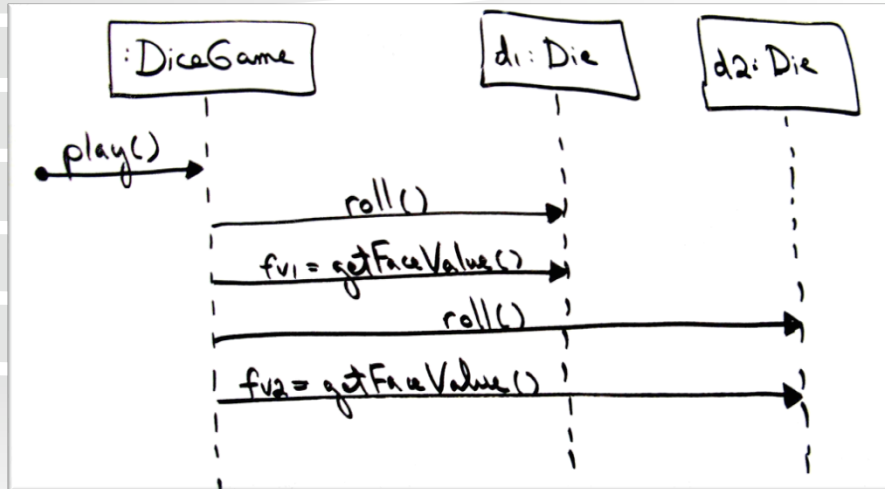
❖ Define Design Class Diagrams



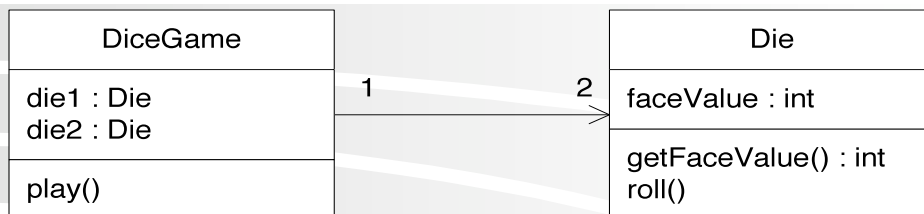
Domain Model for a Dice Game



Sequence diagram for Play Dice Game

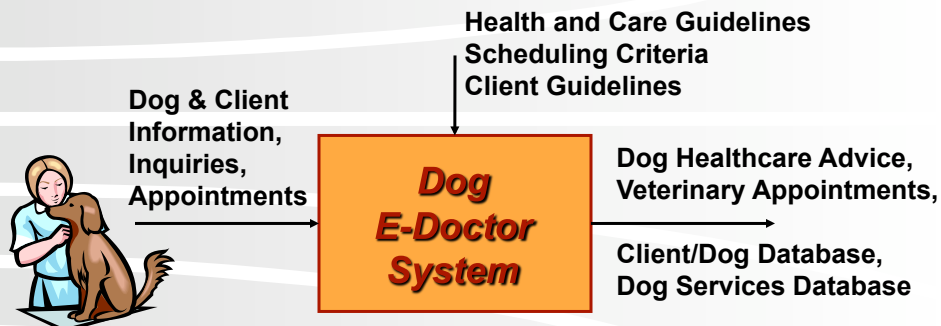


Design Class Diagram for Dice Game



How does it differ from domain model?

Exercise: Dog E-Doctor System



Context: Dog Owners who require health advice or veterinary services for their dog(s).

Purpose: To provide convenient dog healthcare advice and service appointments via the web.

1. Read the problem scope and use case on the handout
2. Answer the quiz questions

Q8, 9

ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Unified Modeling Language (UML)

Grady Booch



Jim Rumbaugh



Ivar Jacobson



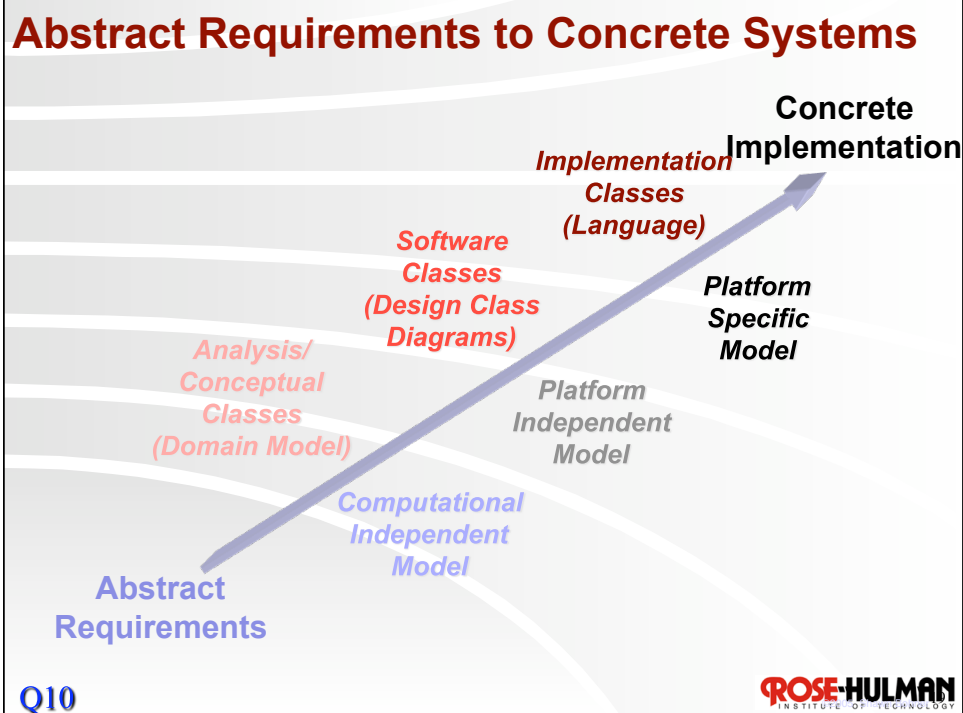
ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

Three Ways to Apply UML

- ❖ Sketch
- ❖ Blueprint
- ❖ Executable programming language

Three Perspectives to Apply UML

- ❖ Conceptual perspective
- ❖ Software specification perspective
- ❖ Software implementation perspective



- ## The Case Studies
- ❖ **NextGen Point of Sale (POS) System**
 - ❖ **Monopoly Game**
 - ❖ **The case study is organized in three iterations**
 - **Each iteration conducts analysis and design on the features for that current software release**
- ROSE-HULMAN**
INSTITUTE OF TECHNOLOGY

Homework Assignment for 12/3/09

- ❖ **Read Chapters 9 through page 148**
- ❖ **Complete Pre-Course Assessment**
 - **Due by 5:00pm on Tuesday, December 1st, 2009**
- ❖ **Milestone 1**
 - **Due by 11:59pm on Friday, December 4th, 2009**