

**W4156**

**Design I: Communicating  
Designs & UML**

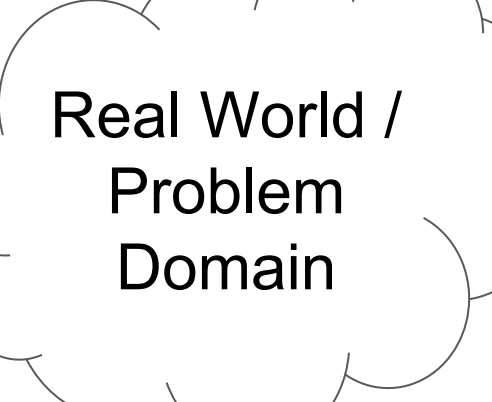
# Agenda

- ❑ Design
- ❑ Why communicate designs?
- ❑ How to communicate a design?
- ❑ UML

# Design (Specifically OOAD)

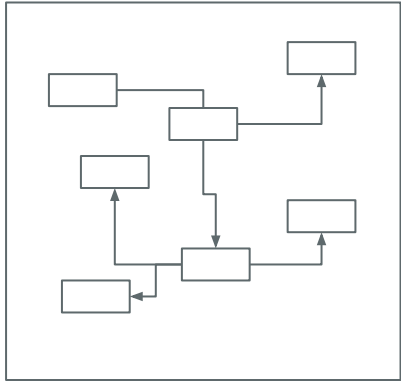
We are on to our next major engineering discipline: design

**Design:** translating our problem domain into components/classes, relationships, methods and data in a way which satisfies functional and non-functional requirements



**Real World /  
Problem  
Domain**

## ➤ Design ➤



# Lecture Arc

## 1: Communicating Designs

How do I communicate the design of a software design/architecture?

## 2: Object Oriented Analysis and Design

What is OOA and how does it help translate a problem domain into a technology solution?

## 3: Patterns

Are there recurring problems and a 'library' of customizable 'good' solutions?

Can I avoid solving every solution from scratch?

**Communicating Designs**

**Volunteers?**

# Ornithologist

- Human
- Earth's top ornithologist (bird expert)
- Never left earth

# Alien

- Extra terrestrial
- Speaks english
- Never visited Earth
- Never seen a bird
- Has 3D printer
- Wants to make a bird

Ornithologist to provide Alien  
Instructions to print ***fully functional*** birds wing

(essentially to convey design of complex system)  
(example taken from Krutchen: [Documenting Software Architectures](#))



# Ornithologist



# Alien





**What did we learn?**

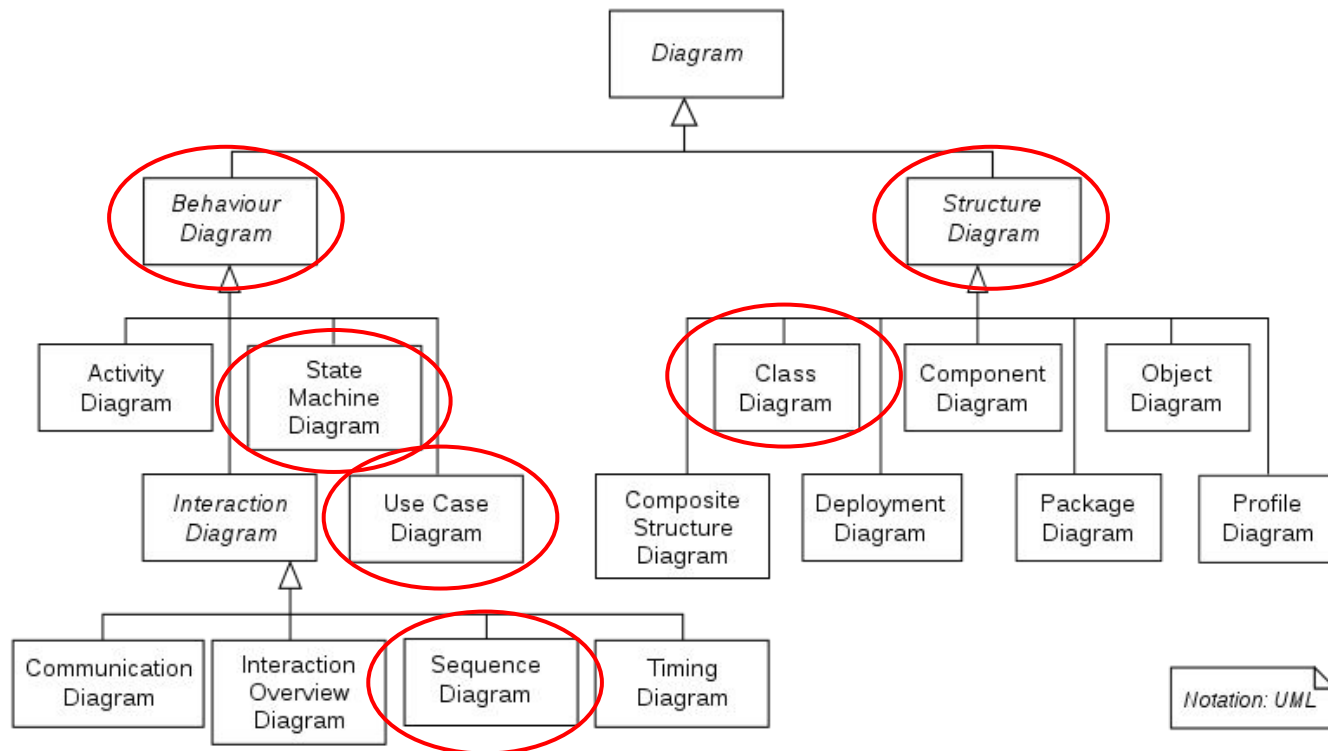
# What did we learn?

1. Conveying design of complex system is hard
2. We need multiple 'views'
3. We need to consider both ***structure and behavior***
4. We need to describe the behavior for different ***scenarios***
5. Our notation was ambiguous but worked in this example  
(we may want to agree a more precise notation)



**UML: A notation to convey designs**

# UML

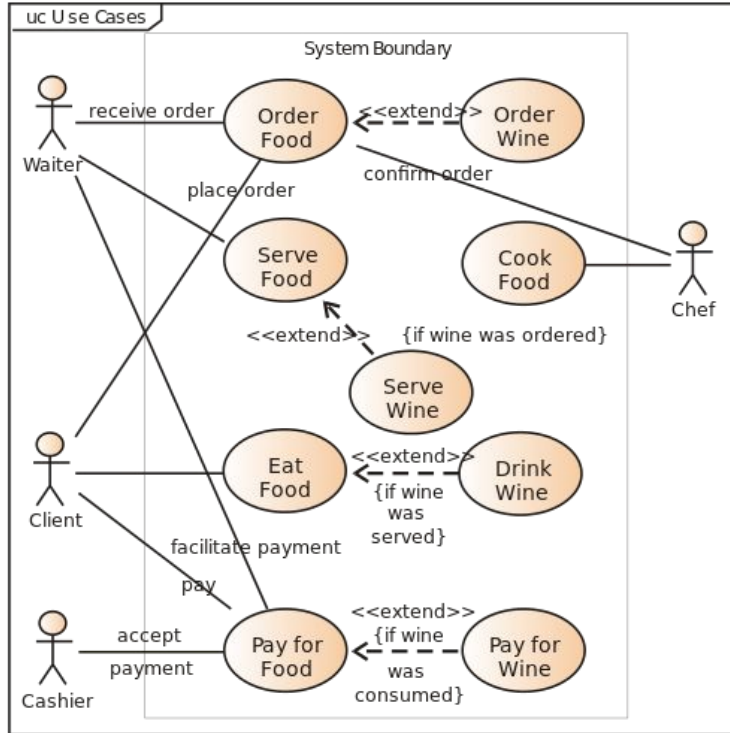


# Use-Case

Title	<Description of the goal the actor is trying to accomplish>
Success Scenario	<Series of steps describing the interaction>
Extensions	<conditions in which the series of steps can differ>

Used to Show	An interaction between an actor (person/system) and a system to achieve a goal
Notation Elements	Use Case Card
Notes	

# Use-Case Diagram



## Used to Show

A use-case shows the interaction of a user with a system.

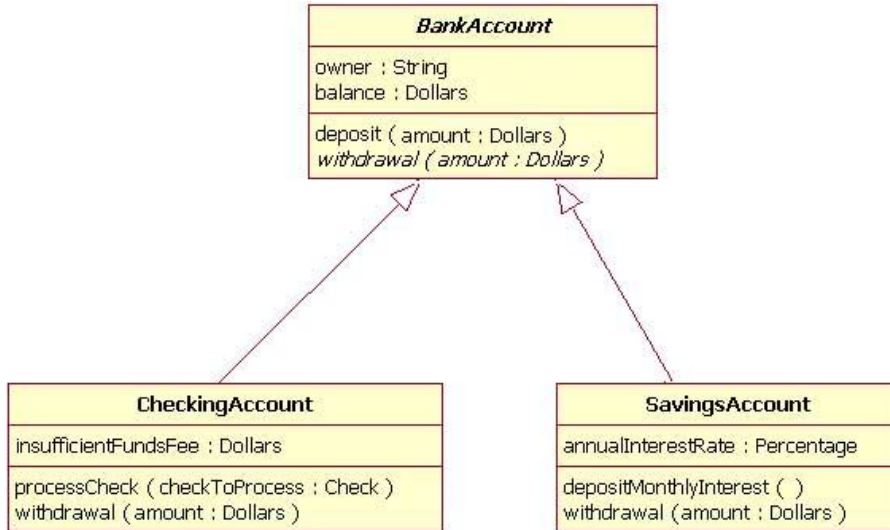
A use-case diagram summarizes the set of use-cases and the relationship between them

## Notation Elements

- Association between actors and use-cases
- Relationships between use-cases (extends, includes)

## Notes




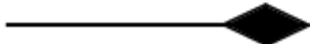
# Class Diagram



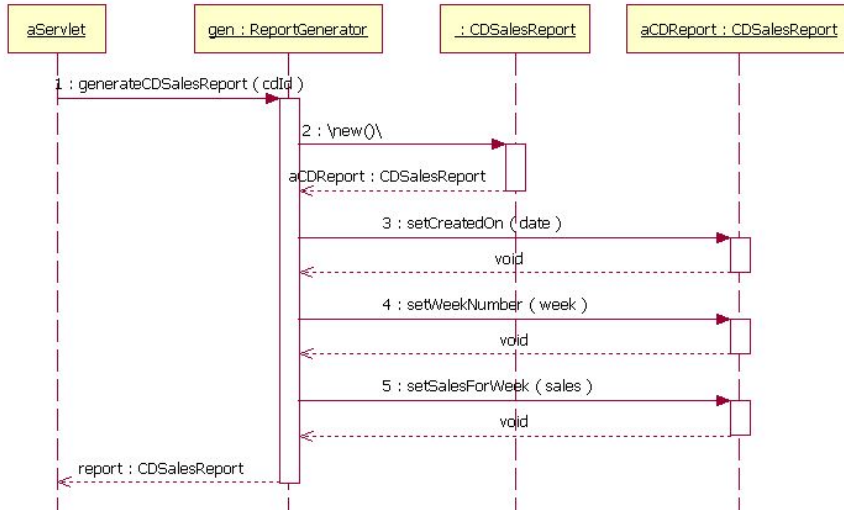
Used to Show	The structural components of a system and the <i>relationship</i> between them (building blocks)
Notation Elements	<ul style="list-style-type: none"><li>- Classes</li><li>- Relationships between classes</li><li>- Multiplicity</li></ul>
Notes	<ul style="list-style-type: none"><li>- Class diagrams can appear two levels of abstraction (conceptual and concrete)</li></ul>



# Class Diagram Associations

Type	Description	Example	Icon
Association	Uses	Car uses road	
Inheritance	Is a	Professor is a human being	
Aggregation	Weak containment	Grouping. Child can exist independent. Delete class and students still exist!	
Composition	Strong containment	Real world whole part. Child not independent. Delete house deletes rooms!	

# Sequence Diagram



<https://www.ibm.com/developerworks/rational/library/769.html>

Used to Show	The the behavior
Notation Elements	<ul style="list-style-type: none"><li>- Lifelines (vertical bars)</li><li>- Extensive notation elements for messages, asynch, sync, loop, if, etc</li></ul>
Notes	<p>Two important things to note</p> <ol style="list-style-type: none"><li>1. A sequence diagram should generally correspond to a use-case showing how that use-case is satisfied</li><li>2. Should be paired with a structural diagram (class/component) The <b>lifelines</b> should correspond to the <b>elements</b> from the structural diagram (class or component)</li></ol>

# Applying UML

# Can we apply?

Can we design software controller for a skittle sorter?

# Pop Quiz

Question	Answer
What is important is {the notation, content}	
What is the difference between structural and behavioral?	
Describe key usage of use-case, class and sequence diagram?	
What is the relationship between UML and Viewpoints and Views?	

# Reading

Reading	Optionality
<u>Springer OOAD</u> (chapters 1,2 & 3)	Required (spread over design lectures)
<u>Springer UML Chapter 8</u>	Required