AI INSTITUTE #AIISC
UNIVERSITY OF SOUTH CAROLINA

UNIVERSITY OF SOUTH CAROLINA

CSCE 240: Advanced Programming Techniques
Lecture 7: Object Oriented Concepts, UML

PROF. BIPLAV SRIVASTAVA, AI INSTITUTE

1ST FEBRUARY 2022

Carolinian Creed: "I will practice personal and academic integrity."

Credits: Some material reused with permission of Dr. Jeremy Lewis. Others used as cited with thanks.

Organization of Lecture 7

- Introduction Section
  - Recap of Lecture 6
  - TA and SI Updates
- Main Section
  - Concept: UML
  - Concept: Object methods
  - Concept: Encapsulation and restriction to access
  - Background for project: Chatbots
- Concluding Section
  - About next lecture – Lecture 8
  - Ask me anything

Introduction Section

Recap of Lecture 6

- We experienced peer review on home works #2
- Discussed objects v/s procedural view of problems
- Introduced Classes/ Objects

Updates from TA, SU

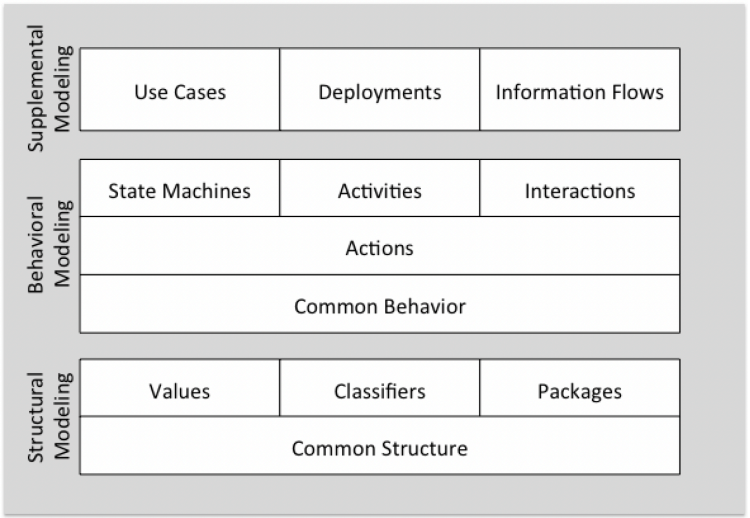
- TA update: Yuxiang Sun (Cherry)
- SI update: Blake Seekings

Main Section

Concept:
Unified Modeling Language (UML)

UML – What is it ?

- A visual, programming language independent, notation for communicating information about an Object-Oriented software's (static – structure, and dynamic - behavior) information
- Latest standard: https://www.omg.org/spec/UML/2.5.1/About-UML/
- Standardized
  - Object Management Group (OMG) adopted in 1997
  - International Organization for Standardization (ISO) published UML as an approved standard in 2005

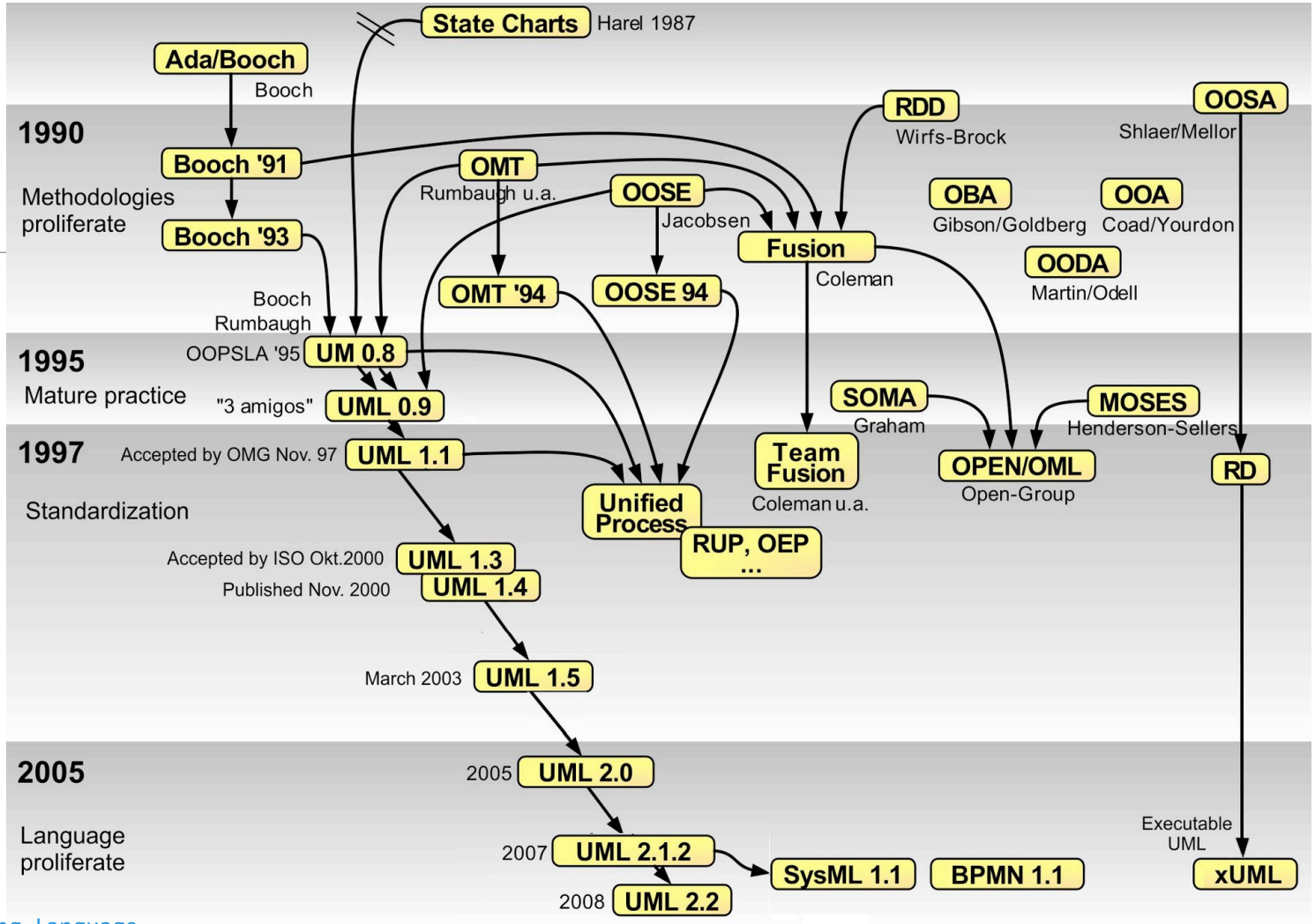


Semantic Areas of UML

Figure credit: UML 2.5.1 Specification

UML History

Rumbaugh, Jacobson and Booch lead the efforts



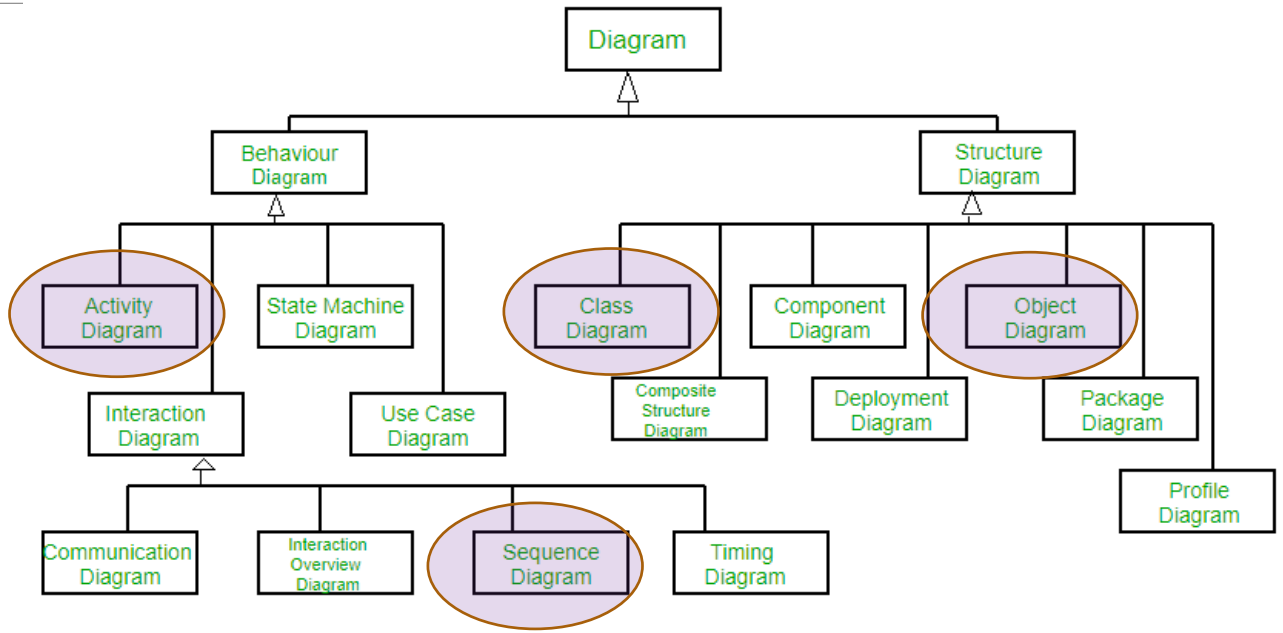
UML History

Figure credit:
https://en.wikipedia.org/wiki/Unified_Modeling_Language

UML – Diagram Types

Most common types

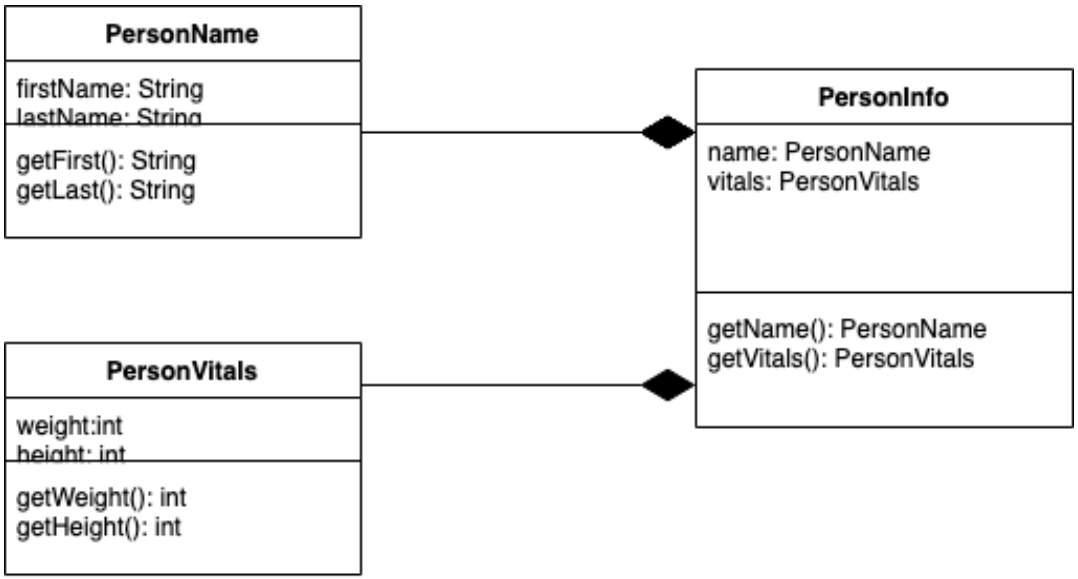
- Class and object diagrams
- Activity or sequence diagram



Types of Diagrams in UML 2.2

Figure credit: https://www.geeksforgeeks.org/unified-modeling-language-uml-introduction/

Example (Modified from Class 5/6)



Relationship Types

- Association: is related to
- Generalization: is a special type of
- Aggregation: is made up of, but can also exist independently
- Composition: is made up of, but cannot exist independently

References:

1. UML 2.5.1 specs
2. Tutorial: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/

Edit Example Class Diagram

- Using browser, go to: https://app.diagrams.net/
- Go to: File -> Open from -> Device -> and load file “Example.drawio”
- Review and edit it
- You can save file or export the diagram in any supported format

Tools for UML

- Many free and paid tools are available
  - See a recent review: https://www.gleek.io/blog/best-uml-tools.html
- We will use diagrams.net (at https://app.diagrams.net/)

Concept: Encapsulation

Encapsulation

- Organize
  - All the information related to a concept together
  - All the methods related to manipulation of the information related to the concept
- Illustration
  - Simple relational number
  - Functions for
    - Accessing data members
    - Utility functions

Reference: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-aggregation-vs-composition/

Encapsulation – Access Restrictions

- Also called visibility rules, applies to both data members and functions
- Types
  - Public: any object can access
  - Protected: only object of same type or children can access
  - Private: only objects of same type can access
- **Demonstration**: SimpleRational

Discussion: Course Project

Course Project – Assembling of Prog. Assignments

- **Project**: Develop collaborative assistants (chatbots) that offer innovative and ethical solutions to real-world problems ! *(Based on competition - https://sites.google.com/view/casy-2-0-track1/contest)*
- Specifically, **the project will be building a chatbot that can answer questions about a South Carolina member of state legislature from**: https://www.scstatehouse.gov/member.php?chamber=H
  - Each student will choose a district (from 122 available).
  - Programming assignment programs will: (1) extract data from the district, (2) process it, (3) make content available in a command-line interface, (4) handle any user query and (5) report on interaction statistics.

Chatbots - Background

- Conversation agents and interfaces (chatbots) are getting easy to build and deploy
  - Can be text-based or speech-based
  - Usually multi-modal (i.e, involving text, speech, vision, document, maps)
- Current chatbots typically interact with a single user at a time and conduct
  - Informal conversation, or
  - Task-oriented activities like answer a user's questions or provide recommendations

Demonstrations

- *Eliza*, http://www.manifestation.com/neurotoys/eliza.php3
- *Mitsuku*, https://www.pandorabots.com/mitsuku/

Chatbots and Their Complexity

- **Users**: 1 or more; stay same or change over time
- **Modality**: variety of input data
- **Data:** static, changes, e.g. sensor data
- **Personalization**: Language, communication style, ..
- **Form**: interface variety
- **Purpose**: what does it help with?
- **Domains**: scope

S.No.	Dimension	Variety
1	User	1, multiple
2	Modality	only conversation, only speech, multi-modal (with point, map, ...)
3	Data source	none, static, dynamic
4	Personalized	no, yes
5	Form	virtual agent, physical device, robot
6	Purpose	socialize, goal: information seeker, goal: action delegate
7	Domains	general, health, water, traffic, ...

Discussion: Nature and Simplifications

- Once you select a district, the elected legislator is fixed.
- Some simplifications
  - **Download local copy** v/s web query
  - **Read static content first**
  - **Handle a subset of content**
  - **Have default handling for questions** the chatbot does not understand
- Do project in a language you are most comfortable with
- Use all advanced programming concepts to simplify coding

Suggested Scope is a Drastic Simplification

- **Users**: 1
- **Modality**: text
- **Data:** static (optionally: dynamic – voting history)
- **Personalization**: none
- **Form**: command line
- **Purpose**: information provider
- **Domain**: specific to representative and district

Concluding Section

Lecture 7: Concluding Comments

- We introduced UML – a language independent notation for communicating about OO software
- Looked at concept of encapsulation
- Discussed background of chatbot

About Next Lecture – Lecture 8

Lecture 8: Object Oriented Continued, UML Notations

- Code organization for OO project
- Larger OO examples
- Project – Prog assignment 1 discussion