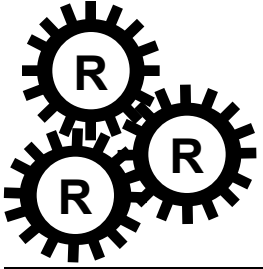


Unified Modeling Language

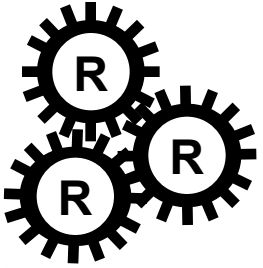
(A Brief Overview)



Types of Diagrams

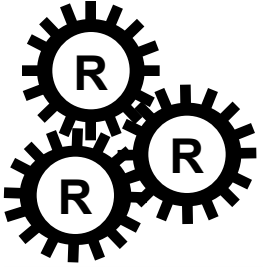
Objectives: visualize, specify, construct, and document a system

- **Structural**: focus on static aspects of system
- **Behavioral**: focus on dynamic aspects of system (changing parts)



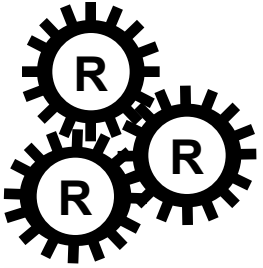
Structural Diagrams

- **Class**: set of classes and their relationships
 - Interface: is a collection of operations that specify a service of a class
- **Object**: set of objects and their relationships
- **Component**: set of components and their relationships
 - component: physical realization of a logical grouping of elements (e.g., classes, interfaces)
- **Deployment**: set of nodes and their relationships
 - exists at run time; represents computational resource
 - node typically encloses one or more components
- **Package**: collection of related UML entities (e.g., classes, interfaces), put together for readability/abstraction



Behavioral Diagrams

- **Use case**: organize behaviors of system
 - user goals (high-level services of system)
 - perspective from external entities (actors)
- **Interaction Diagrams**
 - **Sequence**: focus on time ordering of messages
 - **Collaboration**: focus on structural organization of objects that send/receive messages
- **Statechart**: changing state of system driven by events
- **Activity**: focus on flow of control from one activity to another



Development Process

- High-Level capture of requirements
 - Use Case Diagram
- Identify major objects and relationships
 - Class diagram (object diagram)
- Create scenarios of usage
 - Interaction Diagrams
 - Sequence Diagram
 - Collaboration Diagram
- Generalize scenarios to describe behavior
 - State Diagram
 - Activity Diagram
- Refine to add implementation details
 - Implementation Diagrams
 - Component Diagram
 - Deployment Diagram