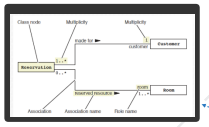

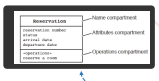

Unified Modeling Language

- Examining the Benefits of Modeling Software
  - Not all developers are involved in each workflow.
    - Example: Sometimes only the Project Manajer and the System Architect will gather requerriments
  - Document the decisions made in heach OOSD workflow
  - Communicate decisions to the project stakeholders
    - Some models can be easily understood by the business owner, domain experts, and other client side stakeholders
  - Modeling enables:
    - Helps to understand, conceptualize and visualice any kind of systems.
    - Visualize new or existing systems
  - Specify the structure (static) and behavior(dynamic) elements of a System
    - Design and Implementation workflows, the structural and dynamic aspects are specified in the Requirements model
    - Solution model represents the complete conceptualization
      - This model is used as a template for implementing the code modules
  - Different views show the model from different perspectives
    - A software model can also be represented by different views, and you can create diagrams for all these views
    - Why Model Software?
      - Building models can facilities the understanding of the system we are developing
  - The development team must create a series of conceptual models that transform a idea into a production system.
  - From architecture:
    - A model is an abstract conceptualization of some entity
    - A model is abstract by its nature
    - Can represent many different things
      - Physical
      - Conceptual
- Exploring the Design Workflow
  - What is a Model?
    - A simplification of reality
    - A descripcion of static and/or dynamic characteristic of a subject area
- Class Diagrams
  - Elements of a Class diagram
  - Associations: represents relationships between classes
  - Multiplicity: determines how many objects might participate in the relationship
  - Navigation: determine what direction an association can be traversed at runtime
- What UML is and is not
  - UML is not:
    - used to create an executable model
    - programming language
    - development process
  - UML it is:
    - generate code skeletons
    - maps the most OO languages
    - enable a software designer to represent programming language constructs
  - UML modeling tools:
    - Provides computer-aided drawing of UML diagrams
    - Support semantic verifications of the diagrams
    - Provide support for a specific methodology
    - Organize the diagrams for a project
    - Automatic generation of modeling elements for design patterns
  - Workflows (disciplines) define the activities that transform the artifacts
  - The OOSD process starts with gathering the system requirements and ends with deploying a working system.
- Elements and Connectors
  - Package: is used to group together any UML elements
  - Note: additional information are attached to any UML element
  - Note: allows textual notes that are added to any aspect of a diagram
  - Dependency: shows that one UML element depends on another UML element
  - Stereotypes: declare a more specific type of element or connector
- UML Diagrams
  - Types of diagrams:
    - Interaction overview diagram: represents a form activity diagram where nodes can represents interaction with the others fragments of the diagram
    - Timing diagram: represents changes in state
    - Case diagram: Represents the set of high-level behaviors
    - Class diagram: represents a collection of software classes and their interrelationships
    - Object diagram: represents a runtime snapshot of software objects and their interrelationships
    - Communication diagram: represents a collection of objects that work together to support some system behavior
    - Sequence diagram: represents a time-oriented perspective of an object communication
    - Activity diagram: represents a flow of activities that might be performed by either a system or an actor
    - State Machine diagram: represents the set of states that an object might experience
    - Component diagram: represents a collection of physical software components
    - Composite Structure diagram: represents the internal structure of a classifier, usually in form parts
    - Profile diagram: define additional diagram types or extend existing diagrams with additional notations
  - Diagrams are categorized in two main categories
    - structural: show the static structure of the objects
    - behavior: show the dinamic behavior of objects
- OOSD as Model Transformations
  - Defining the UML
    - Unified Modeling Language (UML) is a graphical language for visualizing, specifying, constructing and documenting the artifacts of a software system
    - UML is composed of:
      - Elements
      - Diagrams
      - Views
    - Diagrams are built from modeling primitives or elements
    - two broad categories of elements:
      - things (called nodes)
      - relationships (called links)
    - UML diagrams create a visualizations of your mental models
    - these diagrams are used to construct many of the artifacts
  - Functional requirements of Requeriments model
    - merge with the Architecture model to produce the Solution Model
      - defines the details of the structure of the software solution
      - is used to guide the construction of the code
    - Design Model
      - defines the abstract components of the software solution