Unified Modeling Language

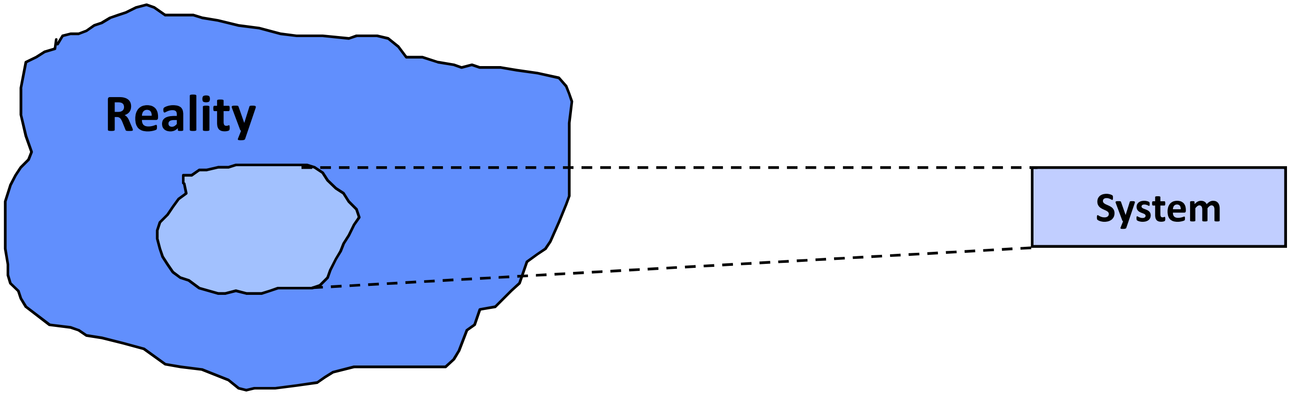
Introduction to the modeling world

Introduction

- UML or Unified Modeling Language comes from Rumbaugh, Booch, and Jacobson (the three amigos) who combined efforts to standardize on one modeling language
- This is primarily a graphical communication mechanism for developers and customers
- We will learn some, but not all, of the UML
  - it is very complex, few understand all of it

What is a model ?

- A model is a simplification of reality.
- Model provides a blueprint of a system
- When you make **a model** you are making a **mapping from the problem domain to a representation** of the system you are modeling.



Principles of modeling

- Principle 1: " The choice of what models to create has a profound influence on how the problem is attacked and the solution is shaped. "
- Principle 2: " Every model may be expressed at different level of precision. "
- Principle 3: " The best models are connected to reality. "
- Principle 4: " No single model is sufficient. "

Why UML ?

- UML is a Language for
  - Visualizing
  - Specifying
  - Constructing
  - Documenting

UML is a Language

- A language provides a vocabulary and some rules for combining words in the vocabulary.
- The vocabulary and rules of modeling language focuses on the conceptual and physical representation of a system.
- For modeling language the notations are their vocabulary and there are some predefined rules for using them.

UML is a language for Visualizing

- Most of us when given a programming problem, we just think it and we code it.
- Still we are doing some modeling
  - but mentally
- However there are several problems with this
  - Communication is harder.
  - Hard to reconstruct.
  - Some important property of the s/w can sometimes be skipped.
- Modeling can be
  - Textual
  - Graphical
- Since UML has some well defined notations and semantics so any designer can visualize the system.

UML is a language for Specifying

- Specifying means building a model that is
  - Precise
  - Unambiguous
  - Complete
- UML addresses the specification of all the important decision of
  - Analysis
  - Design
  - Implementation

UML is a language for Constructing

- UML is not a programming language.
- But it can be directly used to construct code in variety of languages.
- UML expresses the things graphically while programming language expresses the things textually.
- Forward engineering : Construction of a code from a model.
- Reverse Engineering : Reconstruction of the model from the code itself.

UML is a language for Documenting

- The following documents should also be maintained by s/w developers
  - Requirement
  - Architecture
  - Design
  - Source code
  - Project plan
  - Tests
  - Prototype
  - Releases

Where can we use UML ?

- Enterprise information system
- Banking and financial services
- Telecommunication
- Transportation
- Defense/ aerospace
- Retail
- Medical electronics
- Scientific
- Distributed web-based services

Conceptual Model

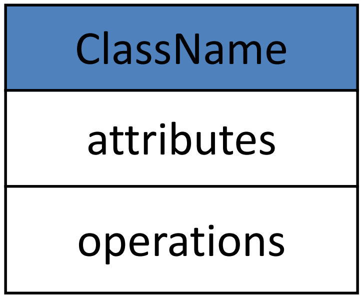
- Building blocks
  - Things
  - Relationships
  - Diagrams
- Things are the abstractions that are the first class citizens in a model.
- Relationship ties things together.
- Diagram groups interesting collection of things.

Things

- Four kinds of things are in UML
  - Structural things
  - Behavioral things
  - Grouping Things
  - Annotational things

Structural Things

- These are the nouns in UML.
- Mostly there are seven kind of structural thing
  - Class - Set of objects sharing same attribute, operations, relationship and semantics.

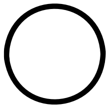


ClassName

attributes

operations

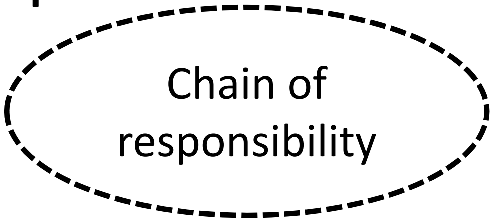
  - Interface- A collection of operations.



Interface

Structural Things

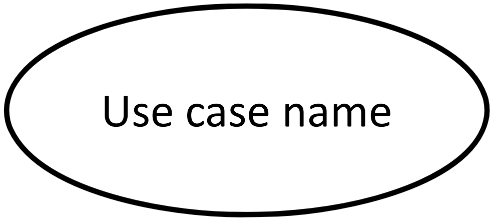
- Collaboration- defines an interaction and a society of roles and other elements that works together to provide cooperative behavior.



Chain of responsibility

Collaborations

- Use case- A description of set of sequence of action.

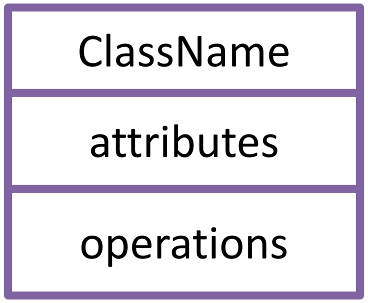


Use case name

Use case

Structural Things

- Active Class- A class whose object owns a process or a thread.



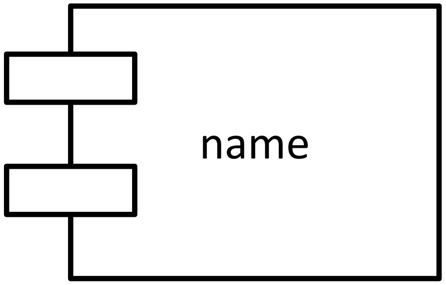
ClassName

attributes

operations

Active Class

- Components- A physical and replaceable part of a system.

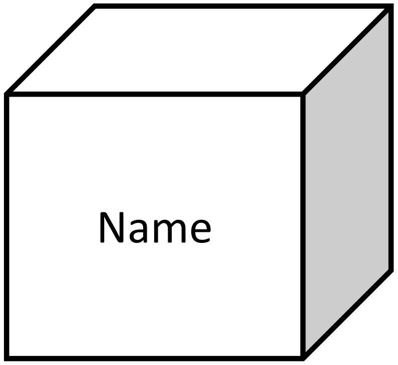


name

Component

Structural Things

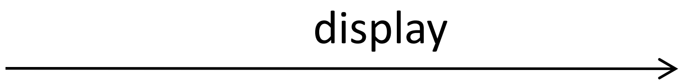
– Server- A node with some memory and processing capability.



Server

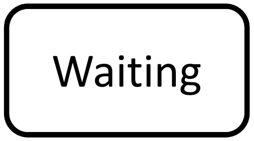
Behavioral Things

- Dynamic part of a model
- Acts as the verb of the model
- Two kinds of behavioral things are present-
  - Interaction - massage, action sequence, links etc.



Message

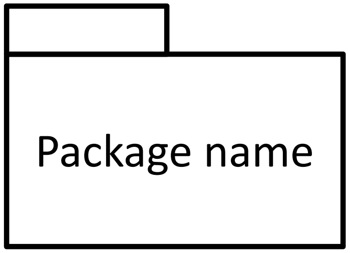
  - State machine- states, events, transitions



States

Grouping Things

- Organizational part of UML.
- One kind of grouping things are available in UML
  - Packages- General purpose mechanism for organizing.



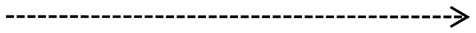
Package

Annotational Things

- Explanatory part of UML.
- Usually notes are used.

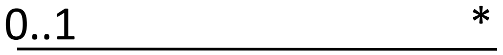
Relationships

- Dependencies- Directed to the things being depended on.



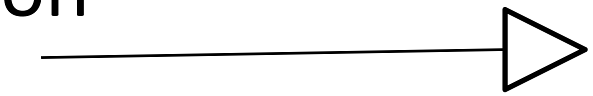
Dependencies

- Association- Connections between objects.



Associations

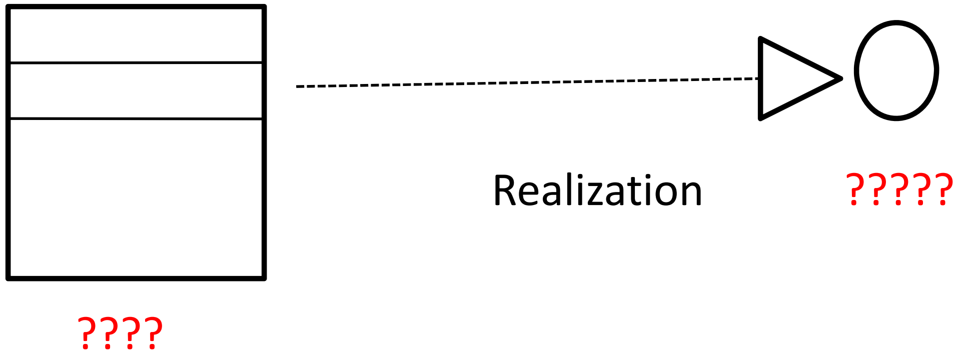
- Generalization-



Generalization

Relationship

- Realization- Used in the context of interface and collaborations.

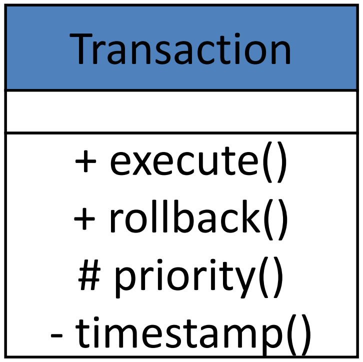


Diagrams

- Class Diagram
- Object Diagram
- Use case Diagram
- Sequence Diagram
- Collaboration Diagram
- State chart Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

Common Mechanism in UML

- adornments :

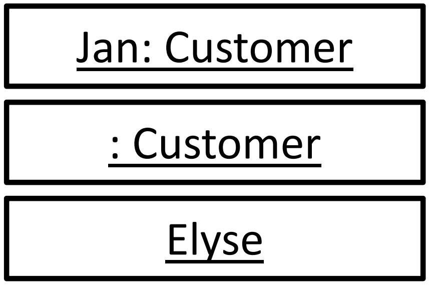


Transaction

+ execute()
+ rollback()
# priority()
- timestamp()

Adornments

- Common Divisions:



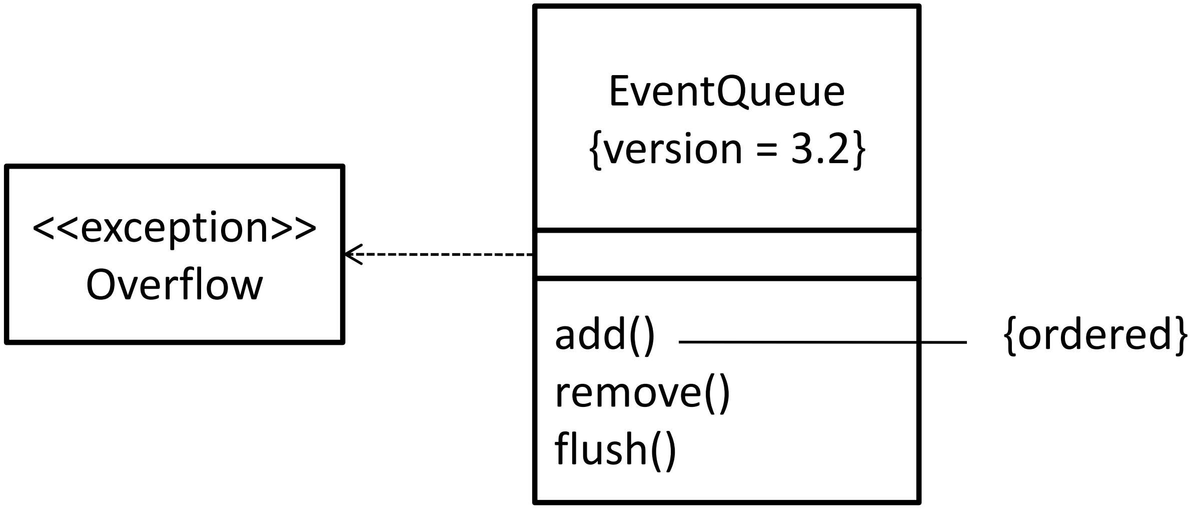
Jan: Customer

: Customer

Elyse

Common Mechanism in UML

- Extensibility Mechanism:
  - Stereotype
  - Tagged Values
  - Constraints



Reference

- UML user Guide
  - Chapter 1,2