Unified Modeling Language

(UML)

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

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 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 (Structural, Behavioral, Grouping, Annotational)
  - Relationships (Dependency, Association, Generalization, Realization)
  - 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 –** Class represents a set of objects having similar responsibilities.

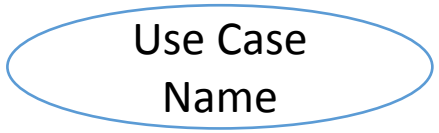
  - **Interface –** Interface defines a set of operations, which specify the responsibility of a class.

Interface

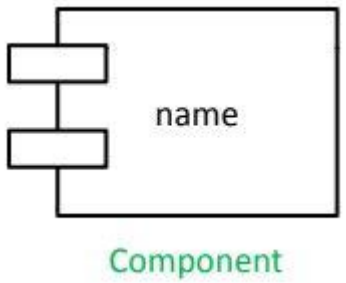
Structural Things

- **Collaboration –** Collaboration defines an interaction between elements.

- **Use case –**Use case represents a set of actions performed by a system for a specific goal.

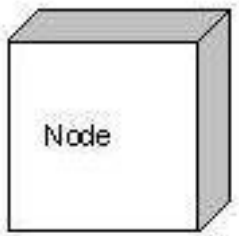


- **Component –**Component describes the physical part of a system.



Structural Things

- **Node/ Server –** A node can be defined as a physical element that exists at run time.



Behavioral Things

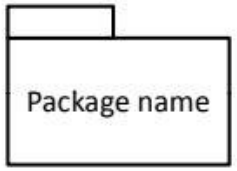
- Dynamic part of a model
- Acts as the verb of the model
- ***Interaction*** *- message, action sequence, links etc.*

- ***State machine*** *- states, events, transitions*

Grouping Things

- *Organizational part of UML.*
- *One kind of grouping things are available in UML*

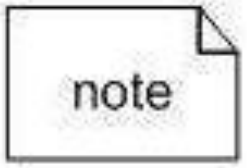
Packages- *General purpose mechanism for organizing.*



Package

Annotational Things

- The explanatory part of the UML model; adds information/meaning to the model elements.
- mechanism to capture remarks, descriptions, and comments of UML model elements.



Relationship

shows how the elements are associated with each other and this association describes the functionality of an application.

- **Dependency**

  a change in one thing (the independent thing) causes a change in the semantics of the other thing (the dependent thing).

- **Association**

  describes the connection between two things.

Relationship

- **Generalization**
  - a relationship between a general thing (called “parent” or “superclass”) and a more specific kind of that thing (called the “child” or “subclass”), such that the latter can substitute the former

(arrow-head points to the superclass)

  - Representing Inheritance
- **Realization**
- a semantic relationship between two things wherein one specifies the behavior to be carried out, and the other carries out the behavior.

(arrow-head points to the thing being realized)

- Mostly found in Interfaces