

UML

The benefits of Modeling Software

Model

A model is a simplification of reality

From architecture a model is an abstract conceptualization of some entity or a system.
A model is abstract by its nature

Why Model Software?

We build models so that we can understand the system we are developing

Models help us understand, conceptualize, and visualize any kind of system.

Communicate decisions to the project stakeholders
Some models can be easily understood by the business owner, domain experts, and other client-side stakeholders.

Specify the structure (static) and behavior (dynamic) elements of a System.

During the Design and Implementation workflows, the structural and dynamic aspects of the software system must be specified from the Requirements model.

The Solution model represents the complete conceptualization of the software system

OOSD(Object Oriented Software Design) as Model Transformations

The project starts as the mental models of the client-side stakeholders.

The non-functional requirements of the Requirements model are transformed in the Architecture model.

The functional requirements of the Requirements model are transformed into a Design model

artifact

- documents
- diagrams
- program

Defining the UML

UML is a graphical language for visualizing, specifying, constructing, and documenting the artifacts

Elements

There are two broad categories of elements:
-things (also called nodes) and
-relationships (also called links).

Diagrams

- Use case
- Class diagram
- Object Diagram
- Communication diagram
- Sequence diagram
- Activity diagram
- State Machine diagram
- Component diagram
- Deployment diagram
- Package diagram
- Interaction Overview
- Timing diagram
- Composite Structure
- Profile

Categories

Structural:
show the static structure of the objects in a system

Behavioral:
show the dynamic behavior of objects in a system

Common UML Elements and Connectors

Package:
-is used to group together any UML elements and diagrams.
-Packages are elements, therefore they can be nested
-is a logical view.

Note:
-a note allows textual notes to be added to any aspect a diagram.
-This allows additional information in the form of text to be attached to any UML element.

Dependency:
-The dependency notation shows that one UML element depends on another UML element.
-The type of dependency can be attached to the line with a stereotype.

Stereotype:
are used to declare a more specific type of element or connector type. The notation for stereotypes is to bracket them in quotation marks symbols « ».

UML modeling tools

- Provide computer-aided drawing of UML diagrams
- Support (or enforce) semantic verification of diagrams
- Provide support for a specific methodology
- Generate code skeletons from the UML diagrams
- Organize all of the diagrams for a project
- Automatic generation of modeling elements for design patterns

Class Diagrams

Associations:
Associations represent relationships between classes. Associations are manifested at runtime in which two objects are associated with each other, usually with an object reference.

Multiplicity:
Multiplicity determines how many objects might participate in the relationship.

Navigation:
Navigation arrows on the association determine what direction an association can be traversed at runtime