Learn UML in 1 Day

By Krishna Rungta

Table Of Content

Chapter 1: UML Diagrams: Versions, Types, History, Tools, Examples

What is UML?

UML stands for **Unified Modeling Language**. It is a standard which is mainly used for creating object-oriented, meaningful documentation models for any software system present in the real world. It provides us a way to develop rich models that describe the working of any software/hardware systems.

UML serves a great way of creating professional documentation which is a necessary part of any project development. UML is an essential part of creating an object-oriented design of systems. It provides you means for creating powerful models and designs for rational systems which can be understood without much difficulties.

Why use UML? Complete History

The 1990s was the era of development of object-oriented languages such as C++. These object-oriented languages were used to create complex but compelling systems.

As the systems developed were complicated to understand, it led tothe design and analysis problems which were faced after the deployment of the system. It was difficult to explain the system to others.

As soon as the UML was introduced, many game-changing experiments and approaches were made for simplifying such difficult tasks of analyzing the system.

UML is an object-oriented unified modeling language. It was invented by brilliant software engineers Grady Booch, Ivar Jacobson, and James Rumbaugh of Rational software during 1994 and 1995. It was under development until 1996.

Each of UML inventors, viz, Grady Booch, Ivar Jacobson, and James Rumbaugh had a fantastic idea for designing a language which will reduce the complexity.

- Booch's method was very flexible to work with during the design and construction of objects.
- Jacobson's method provided a great way to work around use- cases. It also has a powerful approach for high-level design.
- Rumbaugh's method turned out to be very useful while handling sensitive systems.

Later on, behavioral models and state-charts were introduced in the UML which were invented by David Harel.

UML was recognized as a standard by Object Management Group (OMG) during 1997. Object Management Group is responsible for managing UML ever since it was adopted as a standard.

In 2005, the International Organization for Standardization approved UML as an ISO standard. It is used in various industries for creating object-oriented models.

The latest UML version is 2.5.1 which was released in December 2017.

UML Versions

Date	Version	About
November 1997	1.1	UML was adopted by Object Management Group. This was the first version of UML.
March 2000	1.3	A minor upgrade was done to the existing model with notable changes in semantics, notations, and meta-models of UML.
September 2001	1.4	This was the period of the major update to the UML. It scaled UML by providing various extensions. Visibility, artifact, stereotypes were introduced in diagrams.
March 2003	1.5	Features such as procedures, data flow mechanism were added to the UML.
January 2005	1.4.2	UML was accepted as a standard by ISO.
August 2005	2.0	New diagrams such as the object, package, timing, interaction were added to the UML. New features were added to the activity and sequence diagrams. Collaboration diagram was renamed as communication diagram. Multiple features and changes were introduced in the existing diagrams.
April 2006	2.1	Corrections were made to the UML 2.0.
February 2007	2.1.1	Upgrades were introduced in the UML 2.1.
November 2007	2.1.2	UML 2.1.1 was redefined.
February 2009	2.2	UML 2.1.2 bugs were fixed.
May 2010	2.3	UML 2.2 was revised, and minor changes were made to the component diagrams.
August 2011	2.4.1	Classes, packages, and stereotypes changes were made. UML 2.3 was revised with enhancement features.
June 2015	2.5	UML 2.4.1 was revised with minor changes. UML was made simple than it was before. Rapid functioning and the generation of more effective models were introduced. Outdated features were eliminated. Models, templates were eliminated as auxiliary constructs.

Characteristics of UML

1. It is a generalized modeling language.

2. It is different from software programming languages such as Python, C, C++, etc.
3. It is a pictorial language which can be used to generate powerful modeling elements.
4. It is related to object-oriented designs and analysis.
5. It has unlimited applications even outside the software industry. It can be used to visualize the workflow of a factory.

Conceptual model

Before beginning with the UML concept, one must understand the basics of the conceptual model.

A conceptual model is made up of various concepts which are interrelated. It helps us to understand

- What the objects are?
- How interaction takes place to execute a process?

A conceptual model is required in UML. You have to understand the entities and relationships between them before actually modeling the system.

Following object-oriented concepts are required to begin with UML:

- **Object**: It is a real-world entity. There are multiple objects available within a single system. It is a fundamental building block of UML.
- **Class**: A class is nothing but a container where objects and their relationships are maintained.
- **Abstraction**: It is a mechanism of representing an entity without

showing the implementation details. It is used to visualize the behavior of an object.

- **Inheritance**: It is a mechanism of extending an existing class to create a new class.
- **Polymorphism**: It is a mechanism of representing an object having multiple forms which are used for different purposes.
- **Encapsulation**: It is a method of binding the object and the data together as a single unit. It ensures tight coupling between the object and the data.

Above are also called as the **basic building blocks** of a UML.

UML Diagrams

UML diagrams are the output of the Unified Modeling Language. It is a pictorial representation of classes, objects, and relationships between them. UML diagram is a model that describes a part of a system. It is used to define the functionality or a design of a system. A diagram must be clear and concise so that the viewer will readily understand it.

UML diagrams are divided into three different categories such as,

- Structural diagram
- Behavioral diagram
- Interaction diagram

Structural diagrams

Structural diagrams are used to represent a static view of a system. It represents a part of a system that makes up the structure of a system.

A structural diagram shows various objects within the system. Following are

the various structural diagrams in UML:

- Class diagram Object
- diagram Package
- diagram Component
- diagram
- Deployment diagram

Behavioral diagrams

Any real-world system can be represented in either a static form or a dynamic form. A system is said to be complete if it is expressed in both the static and dynamic ways. The behavioral diagram represents the functioning of a system.

UML diagrams that deals with the static part of a system are called structural diagrams. UML diagrams that deals with the moving or dynamic parts of the system are called behavioral diagrams.

Following are the various behavioral diagrams in UML:

- Activity diagram
- Use case diagram
- State machine diagram

Interaction diagrams

Interaction diagram is nothing but a subset of behavioral diagrams. It is used to visualize the flow between various use case elements of a system. Interaction diagrams are used to show an interaction between two entities and how data flows within them.

Following are the various interaction diagrams in UML:

- Timing diagram
- Sequence diagram
- Collaboration diagram

The detailed explanation of the above diagrams is explained in further tutorials.

UML Tools

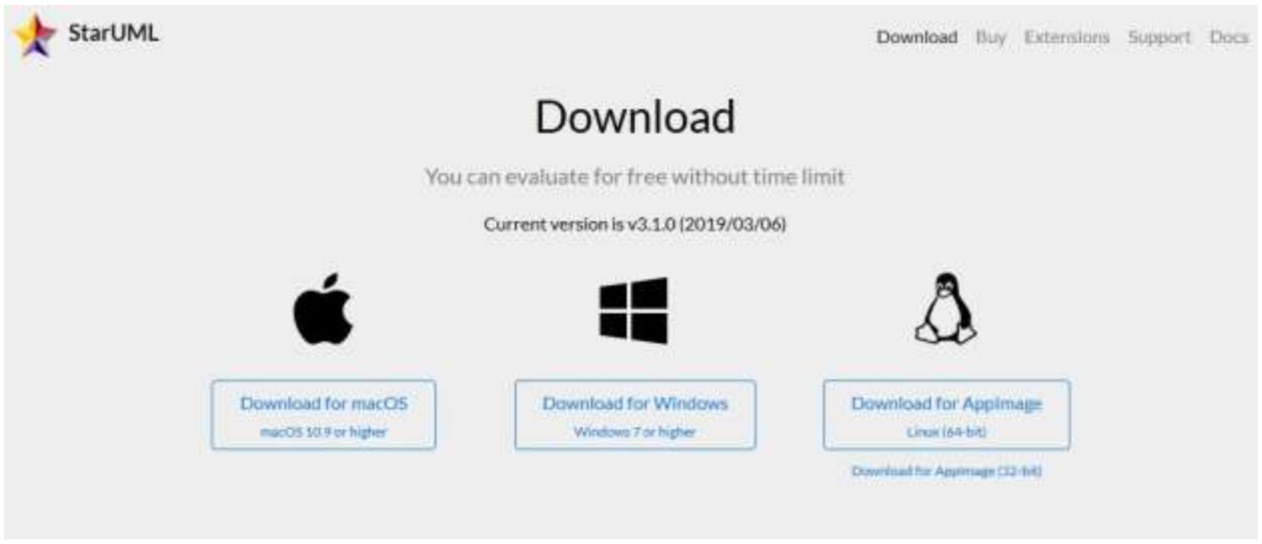
There are many tools available in the market to generate UML diagrams. Some are desktop based while others can be used online. Following is a curated list of tools which can be used for the creation of UML models:

- Star UML
- Argo UML
- Dia
- Visual Paradigm
- U-Model
- UML lab
- Enterprise Architect

We are going to use Star UML application for generating UML diagrams.

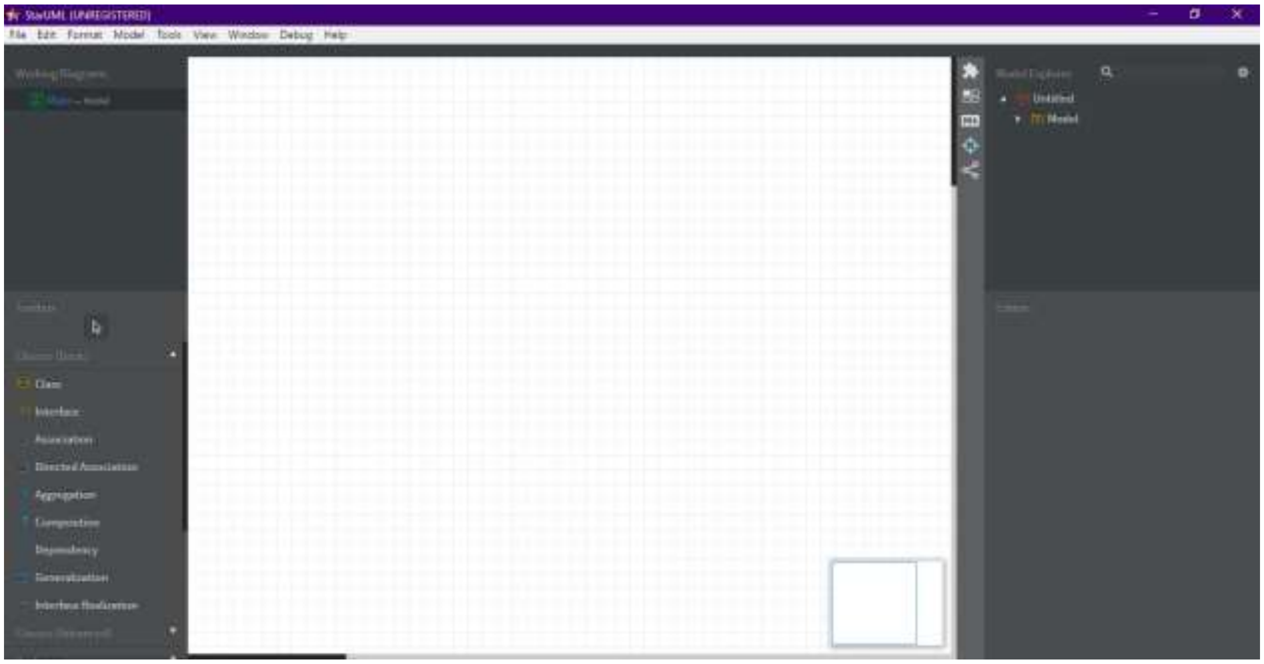
Installation steps: Open the link: http://staruml.io/download

According to your PC specifications. Download any version of the application. Here we are going to choose the windows option.



Once the application is downloaded, install it with all the default options. After installation, launch the Staruml application in your PC.

You will see the following window,



You can now start creating UML diagrams.

Summary

- UML stands for unified modeling language.
- It is used for creating object-oriented models for representing the design and functioning of a system.
- It was developed by Grady Booch, Ivar Jacobson, and James Rumbaugh.
- UML is a successor of object-oriented languages, but it is far different than them.
- Structural, behavioural, and interaction are three types of UML diagrams.
- UML is recognized as an ISO standard which is used by many industries for developing documentation and model blueprints.

Chapter 2: UML Notation Tutorial: Symbol with Examples

What is a model?

A **Model** is an abstraction of something to understand it before building it. As modeling omits unimportant details, it is easier to manipulate than the original entity. A model means organizing something with a particular purpose.

A **model** is a simplification of reality. A

model may provide:

- Blueprint of system
- Organization of the system
- Dynamic of the system

UML Building Blocks

UML stands for unified modeling language which revolves around various blocks to generate a single model. Building blocks are the things required to develop one full UML model diagram. It is an essential part of every UML diagram. Following are the basic UML building blocks:

1. Things

2. Relationships
3. Diagrams

Let us study in depth the building blocks of a UML diagram.

Things

A thing can be described as any real-world entity or an object. Things are divided into various categories in UML as follows,

- Structural things
- Behavioral things
- Grouping things
- Annotational things

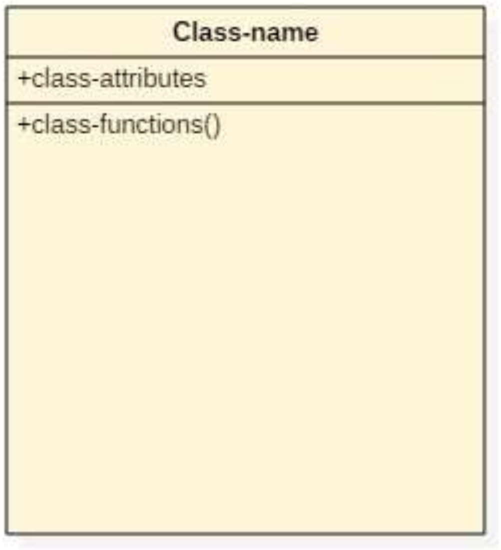
Structural things

A structural thing is used to describe the static part of a model. It is used to represent the things that are visible to human eyes. Structural things are all about the physical part of a system. It is the noun of a UML model, such as a class, object, interface, collaboration, use case, component, and a node.

Structural things consist of:

Class:

A class is used to represent various objects. It is used to define the properties and operations of an object. In UML, we can also represent an abstract class. A class whose functionalities are not defined is called an abstract class. Any UML class notation is generally expressed as follows,

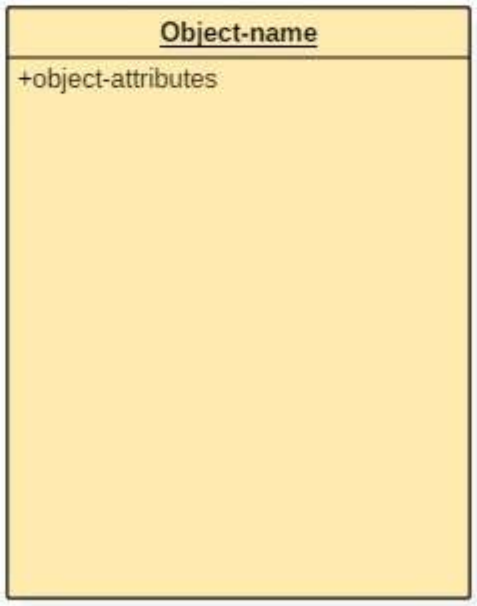


UML Class Symbol

Object:

An object is an entity which is used to describe the behavior and functions of a system. The class and object have the same notations. The only difference is that an object name is always underlined in UML.

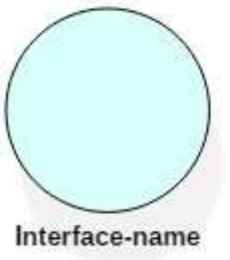
The notation of any object in UML is given below.



UML Object Symbol

Interface:

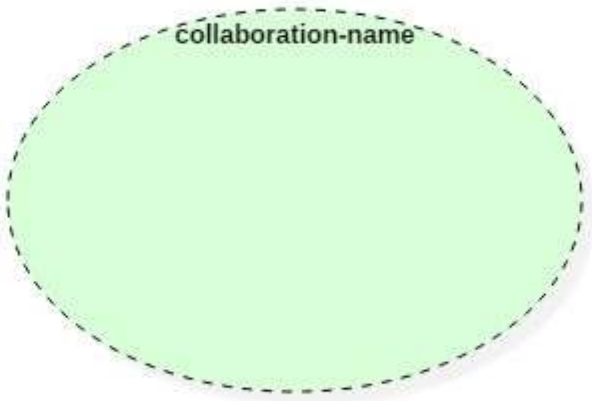
An interface is similar to a template without implementation details. A circle notation represents it. When a class implements an interface, its functionality is also implemented.



UML Interface Symbol

Collaboration:

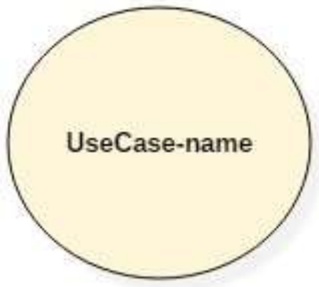
It is represented by a dotted ellipse with a name written inside it.



UML Collaboration Notation

Use-case:

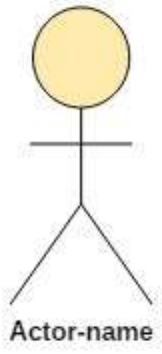
Use-cases are one of the core concepts of object-oriented modeling. They are used to represent high-level functionalities and how the user will handle the system.



UML Use Case

Actor:

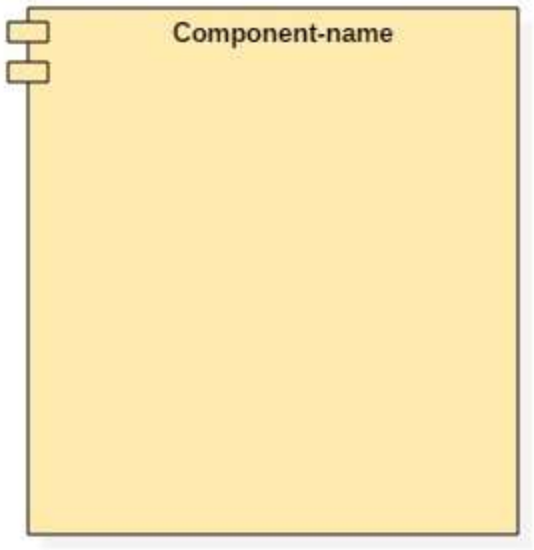
It is used inside use case diagrams. The actor is an entity that interacts with the system. A user is the best example of an actor. The actor notation in UML is given below.



UML Actor

Component:

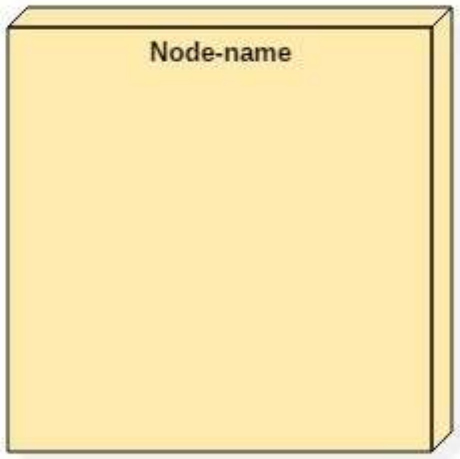
A component notation is used to represent a part of the system. It is denoted in UML like given below,



UML Component

Node:

A node is used to describe the physical part of a system. A node can be used to represent a network, server, routers, etc. Its notation is given below.



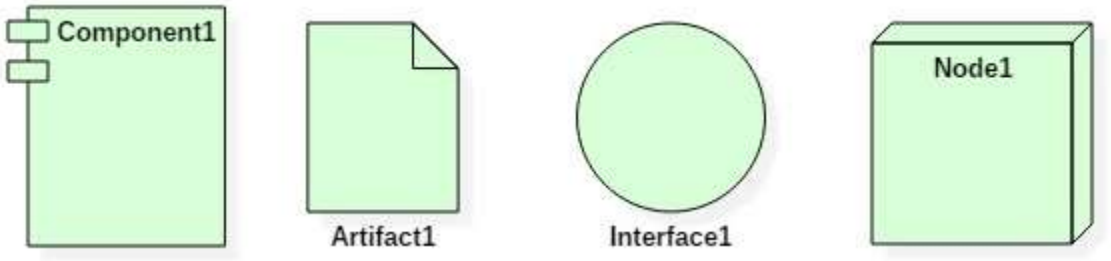
UML Node

Deployment diagram:

It represents the physical hardware on which system is installed. A deployment diagram represents the physical view of a system. It denotes the communication and interaction between various parts of the system.

A deployment diagram consists of the following notations:

1. A node
2. A component
3. An artifact
4. An interface



Deployment Diagram

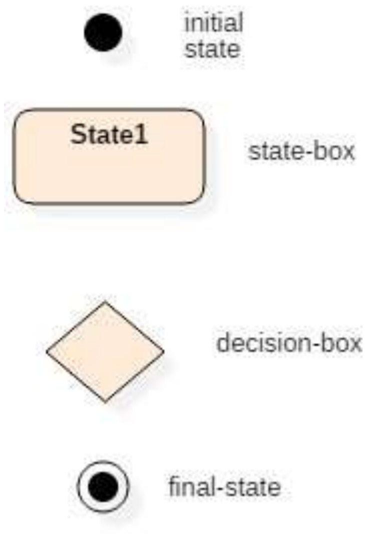
Behavioral things

They are the **verbs** of a UML model, such as interactions, activities and state machines. Behavioral things are used to represent the behavior of a system.

Behavioral things consist of:

State machine:

It used to describe various states of a single component throughout the software development life cycle. It is used to capture different states of a system component.

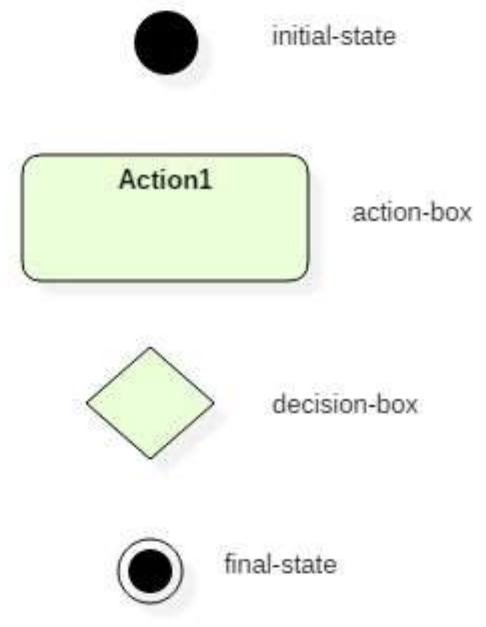


State Machine

Activity diagram:

An activity diagram is used to represent various activities carried out by different components of a system. It is denoted the same as that of the state machine diagram.

Activity diagram mainly contains initial state, final state, a decision box, and an action notation.



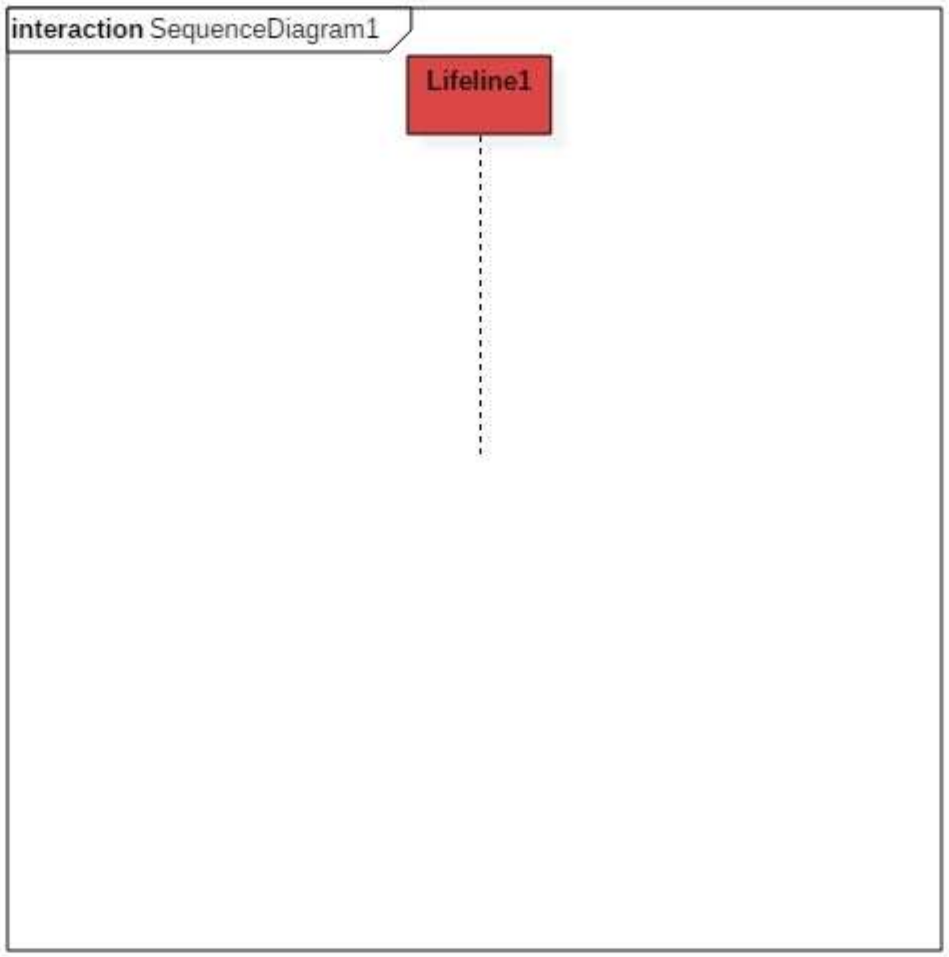
Activity Diagram

Interaction diagram:

Interaction diagrams are used to visualize the message flow between various components of a system.

- Sequence diagram: A sequence diagram shows interactions between one or more lifelines within real time.

The notation of a sequence diagram is given below,



Interaction Diagram