Sequence Diagram::-

What is a Sequence Diagram in UML?

UML Sequence diagrams are interaction diagrams that detail how operations are carried out. As sequence diagrams can be used to capture the interaction between objects in the context of a collaboration, one of the primary uses of sequence diagrams is in the transition from requirements expressed as use cases to the next and more formal level of refinement. Use cases are often refined into one or more sequence diagrams.

Sequence diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

Sequence Diagrams captures interaction in different level of granularity:

- high-level interactions between user of the system and the system, between the system and other systems, or between subsystems (sometimes known as system sequence diagrams)
- the interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- Represent objects interact in (Model, View / Controller) MVC pattern of software framework

Sequence Diagram Notations

Lifeline

A lifeline represents an individual participant in the Interaction.

Actor

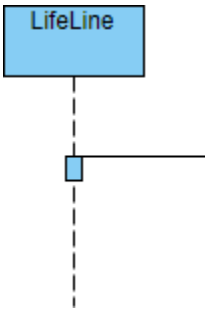
An Actor a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data). An actor can also be an external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). They typically represent roles played by human users, external hardware, or other subjects.

Note That:

- An actor does not necessarily represent a specific physical entity but merely a particular role of some entity
- A person may play the role of several different actors and, conversely, a given actor may be played by multiple different person.

Activation

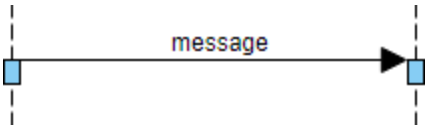
An activation is represented by a thin rectangle on a lifeline) represents the period during which an element is performing an operation. The top and the bottom of the of the rectangle are aligned with the initiation and the completion time respectively



Messages

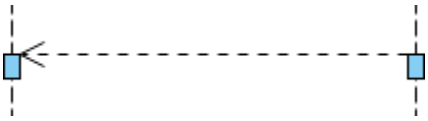
Call Message

A call message defines a particular communication between lifelines of an interaction, which represents an invocation of operation of target lifeline.



Return Message

A return message defines a particular communication between lifelines of an interaction, which represents the pass of information back to the caller of a corresponded former message.

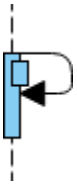


Self Message

A self message defines a particular communication between lifelines of an interaction, which represents the invocation of message of the same lifeline.

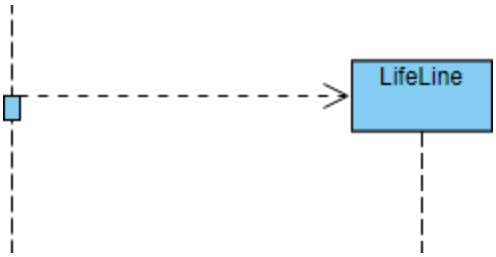
Recursive Message

A recursive message defines a particular communication between lifelines of an interaction, which represents the invocation of message of the same lifeline. It's target points to an activation on top of the activation where the message was invoked from.



Create Message

A create message defines a particular communication between lifelines of an interaction, which represents the instantiation of (target) lifeline.

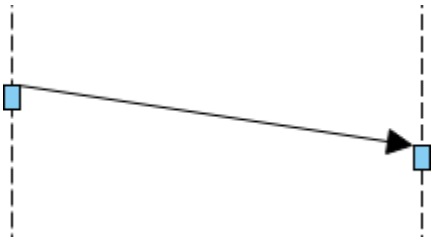


Destroy Message

A destroy message defines a particular communication between lifelines of an interaction, which represents the request of destroying the lifecycle of target lifeline.

Duration Message

A duration message defines a particular communication between lifelines of an interaction, which shows the distance between two time instants for a message invocation.



Note

A note (comment) gives the ability to attach various remarks to elements. A comment carries no semantic force, but may contain information that is useful to a modeler.

When to Draw Sequence Diagram?

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

Sequence Diagram Examples

The sequence diagram example below shows the interactions between a user and a ticket booking system in booking a seat. It consists of mainly four parts: The actor, which is the user, the boundary object ‘interface', the controller object ‘mainController' and two entity objects routes and route.