**Sequence Diagrams**

**Introduction**

Sequence diagrams are a popular dynamic modeling solution in UML because they specifically focus on *lifelines*, or the processes and objects that live simultaneously, and the messages exchanged between them to perform a function before the lifeline ends. Unified Modelling Language (UML) is a modeling language in the field of software engineering which aims to set standard ways to visualize the design of a system. UML guides the creation of multiple types of diagrams such as interaction, structure and behavior diagrams. A sequence diagram is the most commonly used interaction diagram. An interaction diagram is used to show the interactive behavior of a system. Since visualizing the interactions in a system can be a cumbersome task, we use different types of interaction diagrams to capture various features and aspects of interaction in a system. A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

**Benefits of sequence diagrams**

* Represent the details of a UML use case.
* Model the logic of a sophisticated procedure, function, or operation.
* See how objects and components interact with each other to complete a process.
* Plan and understand the detailed functionality of an existing or future scenario.

**Basic Symbols and components**

| Symbol | Name | Description |
| --- | --- | --- |
|  | Object symbol | Represents a class or object in UML. The object symbol demonstrates how an object will behave in the context of the system. Class attributes should not be listed in this shape. |
|  | Activation box | Represents the time needed for an object to complete a task. The longer the task will take, the longer the activation box becomes. |
|  | Actor symbol | Shows entities that interact with or are external to the system. |
|  | Package symbol | Used in UML 2.0 notation to contain interactive elements of the diagram. Also known as a frame, this rectangular shape has a small inner rectangle for labeling the diagram. |
|  | Lifeline symbol | Represents the passage of time as it extends downward. This dashed vertical line shows the sequential events that occur to an object during the charted process. Lifelines may begin with a labeled rectangle shape or an actor symbol. |
|  | Option loop symbol | Used to model if/then scenarios, i.e., a circumstance that will only occur under certain conditions. |
|  | Alternative symbol | Symbolizes a choice (that is usually mutually exclusive) between two or more message sequences. To represent alternatives, use the labeled rectangle shape with a dashed line inside. |

**Common message symbols**

| Symbol | Name | Description |
| --- | --- | --- |
|  | Synchronous message symbol | Represented by a solid line with a solid arrowhead. This symbol is used when a sender must wait for a response to a message before it continues. The diagram should show both the call and the reply. |
|  | Asynchronous message symbol | Represented by a solid line with a lined arrowhead. Asynchronous messages don't require a response before the sender continues. Only the call should be included in the diagram. |
|  | Asynchronous return message symbol | Represented by a dashed line with a lined arrowhead. |
|  | Asynchronous create message symbol | Represented by a dashed line with a lined arrowhead. This message creates a new object. |
|  | Reply message symbol | Represented by a dashed line with a lined arrowhead, these messages are replies to calls. |
|  | Delete message symbol | Represented by a solid line with a solid arrowhead, followed by an X. This message destroys an object. |

**Sequence diagram example**

Here is an example using an emotion based music player:



1. Firstly, the application is opened by the user.
2. The device then gets access to the web cam.
3. The webcam captures the image of the user.
4. The device uses algorithms to detect the face and predict the mood.
5. It then requests database for dictionary of possible moods.
6. The mood is retrieved from the database.
7. The mood is displayed to the user.
8. The music is requested from the database.
9. The playlist is generated and finally shown to the user.

The sequence diagram is a good diagram to use to document a system's requirements and to flush out a system's design. The reason the sequence diagram is so useful is because it shows the interaction logic between the objects in the system in the time order that the interactions take place.

Source:

<https://www.ibm.com/developerworks/rational/library/3101.html>

<https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/>

<https://www.lucidchart.com/pages/uml-sequence-diagram>