

Overview:

UML 2.0 is totally a different dimension in the world of Unified Modeling Language. It is more complex and extensive in nature.

The extent of documentation has also increased compared to UML 1.5 version. UML 2.0 also added new features so that its usage can be more extensive.

UML 2.0 adds the definition of formal and completely defined semantics. This new possibility can be utilized for the development of models and the corresponding systems can be generated from these models. But to utilize this new dimension a considerable effort has to be made to acquire the knowledge.

New dimensions in UML2.0:

The structure and documentation of UML was completely revised in the latest version of UML2.0. There are now two documents available that describe UML:

- UML 2.0 Infrastructure defines the basic constructs of the language on which UML is based. This section is not directly relevant to the users of UML. This is directed more towards the developers of modeling tools. So this area is not in the scope of this tutorial.
- UML 2.0 Superstructure defines the user constructs of UML 2.0. It means those elements of UML that users will use at the immediate level. So this is the main focus for the user community of UML.

This revision of UML was created to fulfil a goal to restructure and refine UML so that usability, implementation, and adaptation are simplified.

The UML infrastructure is used to:

- Provide a reusable meta-language core. This is used to define UML itself.
- Provide mechanisms to adjustment the language.

The UML superstructure is used to:

- Provide better support for component-based development.
- Improve constructs for the specification of architecture.
- Provide better options for the modeling of behaviour.

So the important point to note is the major divisions described above. These divisions are used to increase the usability of UML and define a clear understanding of its usage.

There is another dimension which is already proposed in this new version. It is a proposal for a completely new Object Constraint Language *OCL* and Diagram Interchange. These features all together form the complete UML2.0 package.

Modeling diagrams in UML2.0:

Modeling Interactions:

The interaction diagrams described in UML2.0 is different than the earlier versions. But the basic concept remains same as the earlier version. The major difference is the enhancement and additional features added to the diagrams in UML2.0.

UML2.0 models object interaction in the following four different ways.

- **Sequence diagram** is a time dependent view of the interaction between objects to accomplish a behavioral goal of the system. The time sequence is similar to the earlier version of sequence diagram. An interaction may be designed at any level of abstraction within the system design, from subsystem interactions to instance-level.
- **Communication diagram** is a new name added in UML2.0. A Communication diagram is a structural view of the messaging between objects, taken from the Collaboration diagram concept of UML 1.4 and earlier versions. This can be defined as a modified version of collaboration diagram.
- **Interaction Overview diagram** is also a new addition in UML2.0. An Interaction Overview diagram describes a high-level view of a group of interactions combined into a logic sequence, including flow-control logic to navigate between the interactions.
- **Timing diagram** is also added in UML2.0. It is an optional diagram designed to specify the time constraints on messages sent and received in the course of an interaction.

So from the above description it is important to note that the purposes of all the diagrams are to send/receive messages. Now the handlings of these messages are internal to the objects. So the objects are also having options to receive and send messages, and here comes another important aspect called interface. Now these interfaces are responsible for accepting and sending messages to one another.

So from the above discussion it can be concluded that the interactions in UML2.0 are described in a different way and that is why the new diagram names have come into picture. But if we analyze the new diagrams then it is clear that all the diagrams are created based upon the interaction diagrams described in the earlier versions. The only difference is the additional features added in UML2.0 to make the diagrams more efficient and purpose oriented.

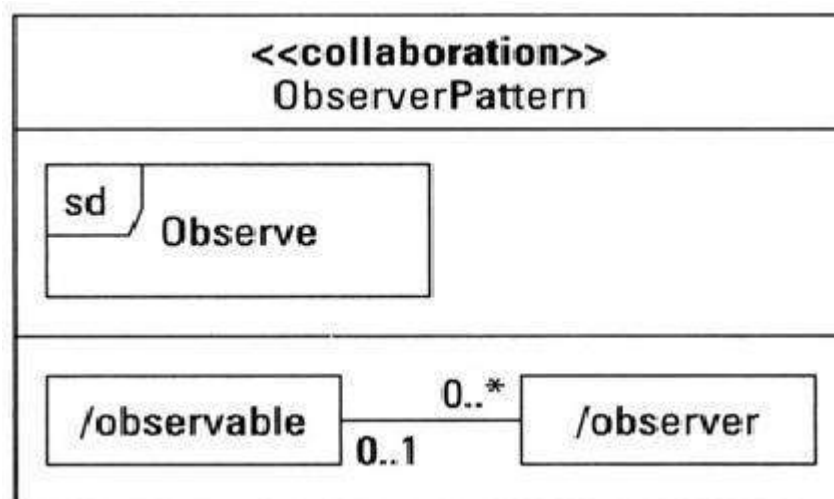
Modeling Collaborations:

As we have already discussed that collaboration is used to model common interactions between objects. To clarify it, we can say that collaboration is a interaction where a set of messages are handled by a set of objects having pre defined roles.

The important point to note is the difference between the collaboration diagram in earlier version and in UML2.0 version. So to distinguish the collaboration diagram the name has been changed in UML2.0. In UML2.0 it is named as Communication diagram.

Consequently collaboration is defined as a class with attributes *properties* and behavior *operations*. Compartments on the collaboration class can user defined also and may be used for interactions *Sequencediagrams* and the structural elements *CompositeStructurediagram*.

Figure below models the Observer design pattern as collaboration between an object in the role of an observable item and any number of objects as the observers.



Modeling Communication:

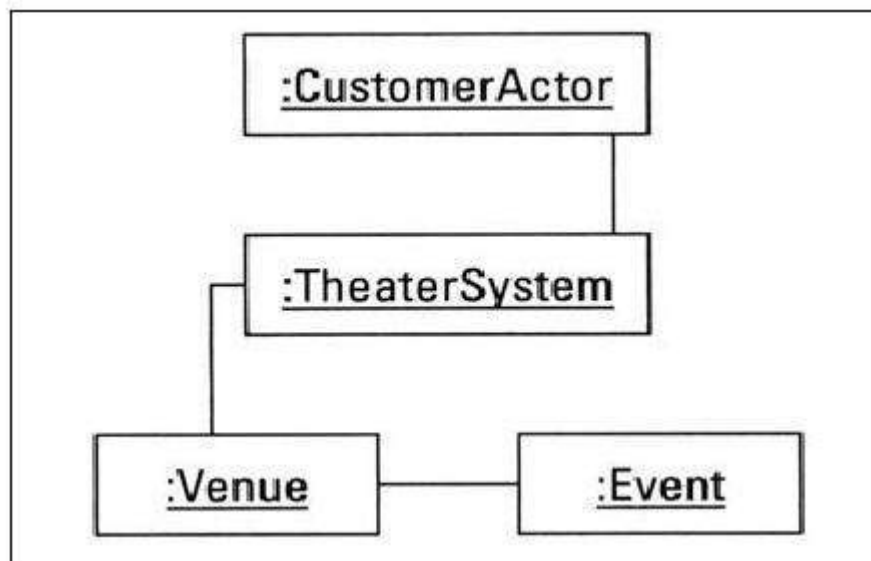
Communication diagram is slightly different than the collaboration diagrams of the earlier versions. We can say it is a scaled back version of the earlier UML versions. The distinguishing factor of the communication diagram is the link between objects.

This is a visual link and it is missing in sequence diagram. In sequence diagram only the messages passed between objects are shown even if there is no link between them.

The Communication diagram is used to prevent the modeler from making this mistake by using an Object diagram format as the basis for messaging. Each object on a Communication diagram is called an object lifeline.

The message types in a Communication diagram are the same as in a Sequence diagram. A Communication diagram may model synchronous, asynchronous, return, lost, found, and object-creation messages.

Figure below shows an Object diagram with three objects and two links that form the basis for the Communication diagram. Each object on a Communication diagram is called an object lifeline.



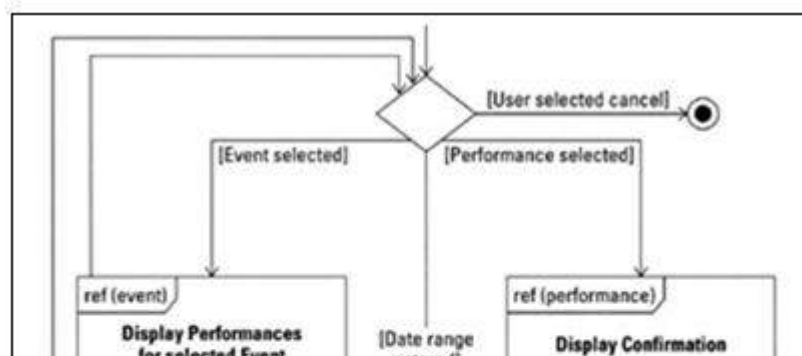
Modeling an Interaction Overview:

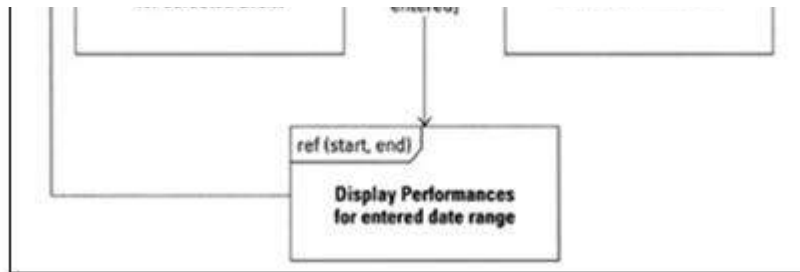
In practical usage, a sequence diagram is used to model a single scenario. So numbers of sequence diagrams are used to complete the entire application. So while modeling a single scenario it is possible to forget the total process and this can introduce errors.

So to solve this issue the new interaction overview diagram combines the flow of control from an activity diagram and messaging specification from the sequence diagram.

Activity diagram uses activities and object flows to describe a process. The Interaction Overview diagram uses interactions and interaction occurrences. The lifelines and messages found in Sequence diagrams appear only within the interactions or interaction occurrences. However, the lifelines *objects* that participate in the Interaction Overview diagram may be listed along with the diagram name.

Figure below shows an interaction overview diagram with decision diamonds, frames and termination point





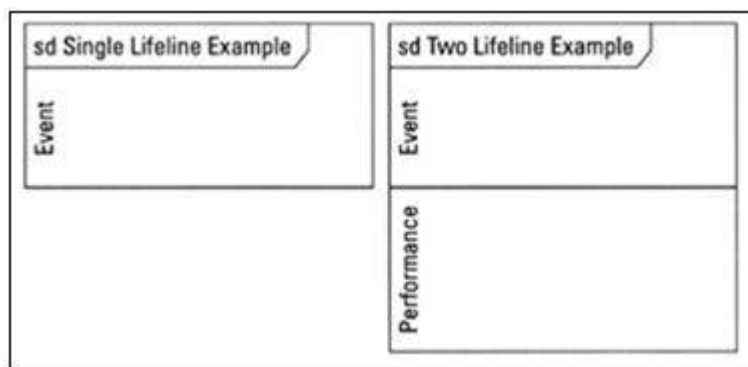
Modeling a Timing Diagram:

The name of this diagram itself describes the purpose of the diagram. It basically deals with the time of the events over its entire lifecycle.

So a timing diagram can be defined as a special purpose interaction diagram made to focus on the events of an object in its life time. It is basically a mixture of state machine and interaction diagram. The timing diagram uses the following time lines:

- State time line
- General value time line

A lifeline in a Timing diagram forms a rectangular space within the content area of a frame. It is typically aligned horizontally to read from left to right. Multiple lifelines may be stacked within the same frame to model the interaction between them.



Summary:

UML2.0 is an enhanced version where the new features are added to make it more usable and efficient. There are two major categories in UML2.0, One is UML super structure and another is UML infrastructure. Although the new diagrams are based on the old concepts but still they have additional features.

UML 2.0 offers four interaction diagrams, the Sequence diagram, Communication diagram, Interaction Overview diagram, and an optional Timing diagram. All four diagrams utilize the frame notation to enclose an interaction. The use of frames supports the reuse of interactions as

interaction occurrences