Different UML Diagrams

Dr. Buddhadeb Pradhan

Deployment Diagrams

- Deployment diagrams are used to visualize the topology of the physical components of a system, where the software components are deployed.
- Deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

Purpose of Deployment Diagrams

- The term Deployment itself describes the purpose of the diagram. Deployment diagrams are used for describing the hardware components, where software components are deployed. Component diagrams and deployment diagrams are closely related.
- Component diagrams are used to describe the components and deployment diagrams shows how they are deployed in hardware.
- UML is mainly designed to focus on the software artifacts of a system. However, these two diagrams are special diagrams used to focus on software and hardware components.
- Most of the UML diagrams are used to handle logical components but deployment diagrams are made to focus on the hardware topology of a system. Deployment diagrams are used by the system engineers.

- ❑ The purpose of deployment diagrams can be described as –
- ❑ Visualize the hardware topology of a system.
- ❑ Describe the hardware components used to deploy software components.
- ❑ Describe the runtime processing nodes.

How to Draw a Deployment Diagram?

- Deployment diagram represents the deployment view of a system. It is related to the component diagram because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware used to deploy the application.
- Deployment diagrams are useful for system engineers. An efficient deployment diagram is very important as it controls the following parameters –

✓ Performance
✓ Scalability
✓ Maintainability
✓ Portability

- Before drawing a deployment diagram, the following artifacts should be identified –

❖ Nodes
❖ Relationships among nodes

Sample Deployment Diagram

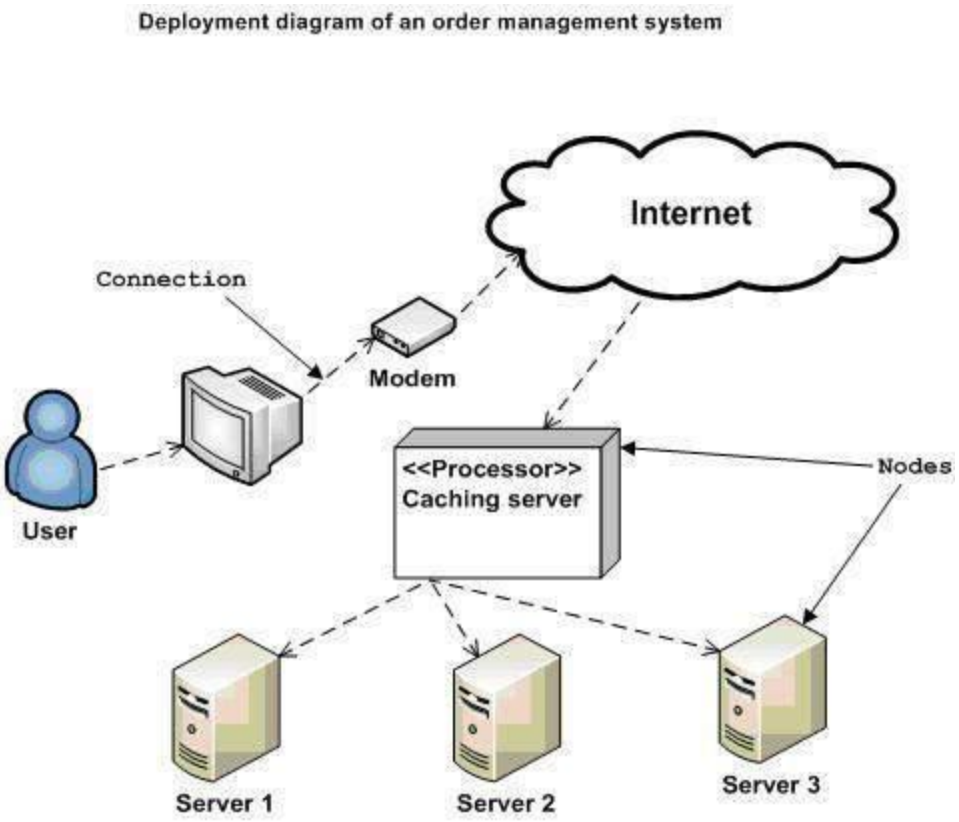
Here is a sample deployment diagram to provide an idea of the deployment view of order management system. Here, we have shown nodes as −

- Monitor
- Modem
- Caching server
- Server

The application is assumed to be a web-based application, which is deployed in a clustered environment using server 1, server 2, and server 3.

The user connects to the application using the Internet.

The control flows from the caching server to the clustered environment.



Where to Use Deployment Diagrams?

- Deployment diagrams are mainly used by system engineers. These diagrams are used to describe the physical components (hardware), their distribution, and association.
- Deployment diagrams can be visualized as the hardware components/nodes on which the software components reside.
- Software applications are developed to model complex business processes. Efficient software applications are not sufficient to meet the business requirements. Business requirements can be described as the need to support the increasing number of users, quick response time, etc.
- To meet these types of requirements, hardware components should be designed efficiently and in a cost-effective way.
- Now-a-days software applications are very complex in nature. Software applications can be standalone, web-based, distributed, mainframe-based and many more. Hence, it is very important to design the hardware components efficiently.

Deployment diagrams can be used –

- ❑ To model the hardware topology of a system.
- ❑ To model the embedded system.
- ❑ To model the hardware details for a client/server system.
- ❑ To model the hardware details of a distributed application.
- ❑ For Forward and Reverse engineering.

Use Case Diagrams

- To model a system, the most important aspect is to capture the dynamic behavior. Dynamic behavior means the behavior of the system when it is running/operating.
- Only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.
- These internal and external agents are known as actors. Use case diagrams consists of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system.
- Hence to model the entire system, a number of use case diagrams are used.

Purpose of Use Case Diagrams

- The purpose of use case diagram is to capture the dynamic aspect of a system. However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from other four diagrams.
- Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.
- When the initial task is complete, use case diagrams are modelled to present the outside view.

In brief, the purposes of use case diagrams can be said to be as follows –

- ❖ Used to gather the requirements of a system.
- ❖ Used to get an outside view of a system.
- ❖ Identify the external and internal factors influencing the system.
- ❖ Show the interaction among the requirements are actors.

How to Draw a Use Case Diagram?

- Use case diagrams are considered for high level requirement analysis of a system. When the requirements of a system are analyzed, the functionalities are captured in use cases.
- We can say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases are the actors. Actors can be defined as something that interacts with the system.
- Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

➢ Functionalities to be represented as use case

➢ Actors

➢ Relationships among the use cases and actors.

How to Draw a Use Case Diagram?

Use case diagrams are drawn to capture the functional requirements of a system. After identifying the above items, we have to use the following guidelines to draw an efficient use case diagram

- The name of a use case is very important. The name should be chosen in such a way so that it can identify the functionalities performed.
- Give a suitable name for actors.
- Show relationships and dependencies clearly in the diagram.
- Do not try to include all types of relationships, as the main purpose of the diagram is to identify the requirements.
- Use notes whenever required to clarify some important points.

Sample Use Case Diagram

Here is a sample use case diagram representing the order management system. Hence, if we look into the diagram then we will find three use cases **(Order, SpecialOrder, and NormalOrder)** and one actor which is the customer.

The SpecialOrder and NormalOrder use cases are extended from *Order* use case. Hence, they have extended relationship.

Another important point is to identify the system boundary, which is shown in the picture.

The actor Customer lies outside the system as it is an external user of the system.

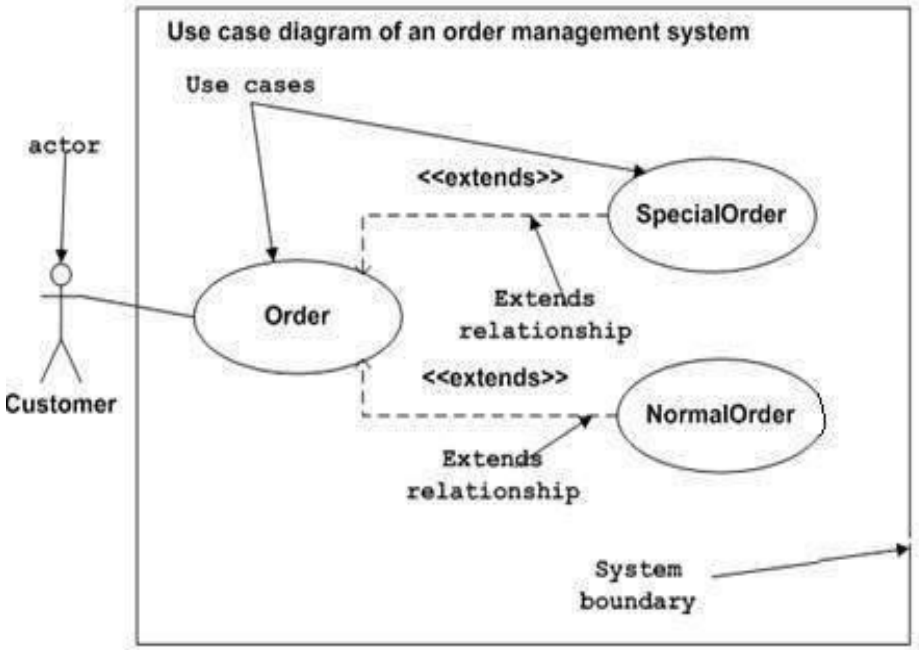
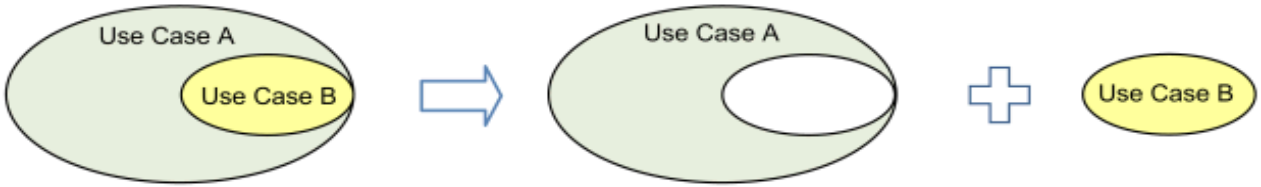
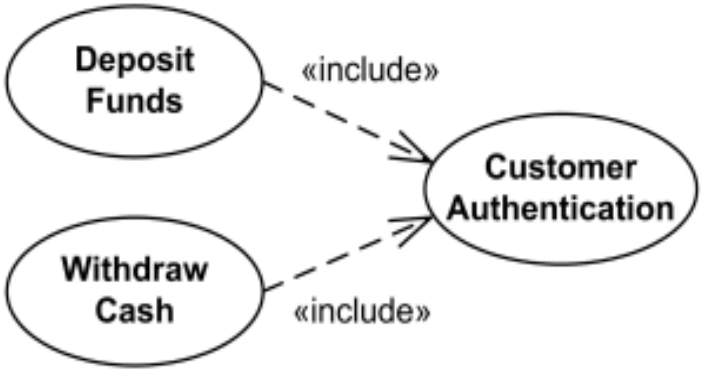


Figure: Sample Use Case diagram

UML Use Case Include



- **Use case include** is a directed relationship between two use cases which is used to show that behavior of the **included** use case (the addition) is inserted into the behavior of the **including** (the base) use case.
- The **include** relationship could be used:
- to simplify large use case by splitting it into several use cases,
- to extract **common parts** of the behaviors of two or more use cases.
- A large use case could have some behaviors which might be detached into distinct smaller use cases to be included back into the base use case using the UML **include** relationship. The purpose of this action is modularization of behaviors, making them more manageable.



Where to Use a Use Case Diagram?

- To understand the dynamics of a system, we need to use different types of diagrams. Use case diagram is one of them and its specific purpose is to gather system requirements and actors.
- Use case diagrams specify the events of a system and their flows. But use case diagram never describes how they are implemented.
- Use case diagram can be imagined as a black box where only the input, output, and the function of the black box is known.
- A well-structured use case also describes the pre-condition, post condition, and exceptions. These extra elements are used to make test cases when performing the testing.
- In forward engineering, use case diagrams are used to make test cases and in reverse engineering use cases are used to prepare the requirement details from the existing application.
- Use case diagrams can be used for –
- Requirement analysis and high level design.
- Model the context of a system.
- Reverse engineering.
- Forward engineering.

UML - Interaction Diagrams

- From the term Interaction, it is clear that the diagram is used to describe some type of interactions among the different elements in the model. This interaction is a part of dynamic behavior of the system.
- This interactive behavior is represented in UML by two diagrams known as **Sequence diagram** and **Collaboration diagram**. The basic purpose of both the diagrams are similar.
- Sequence diagram emphasizes on time sequence of messages and collaboration diagram emphasizes on the structural organization of the objects that send and receive messages.

Purpose of Interaction Diagrams

- The purpose of interaction diagrams is to visualize the interactive behavior of the system. Visualizing the interaction is a difficult task. Hence, the solution is to use different types of models to capture the different aspects of the interaction.
- **Sequence and collaboration** diagrams are used to capture the dynamic nature but from a different angle.

  The purpose of interaction diagram is −

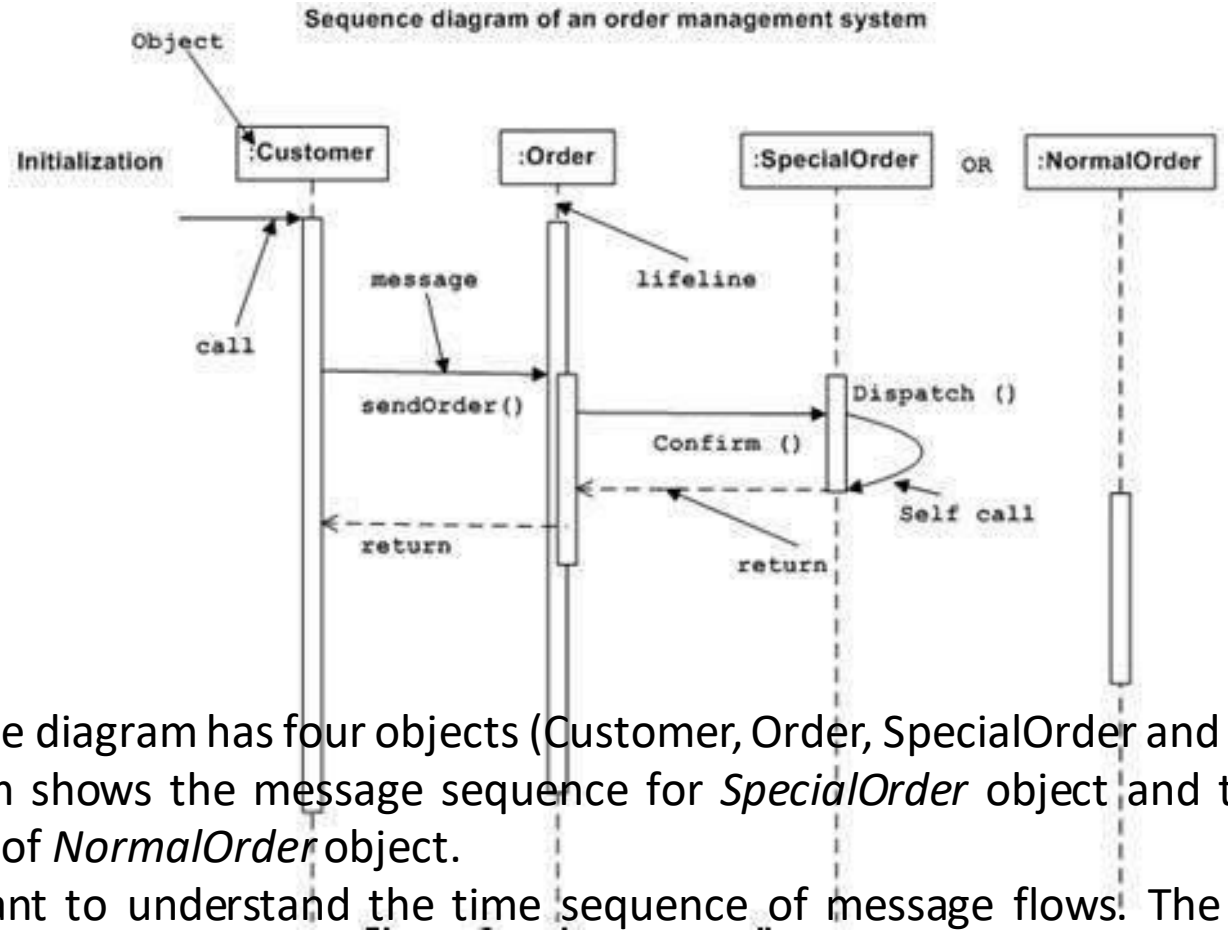
- ❑ To capture the dynamic behaviour of a system.
- ❑ To describe the message flow in the system.
- ❑ To describe the structural organization of the objects.
- ❑ To describe the interaction among objects.

How to Draw an Interaction Diagram?

- The purpose of interaction diagrams is to capture the dynamic aspect of a system. So to capture the dynamic aspect, we need to understand what a dynamic aspect is and how it is visualized. Dynamic aspect can be defined as the snapshot of the running system at a particular moment.
- We have two types of interaction diagrams in UML. One is the sequence diagram and the other is the collaboration diagram. The sequence diagram captures the time sequence of the message flow from one object to another and the collaboration diagram describes the organization of objects in a system taking part in the message flow.
- Following things are to be identified clearly before drawing the interaction diagram

❑ Objects taking part in the interaction.
❑ Message flows among the objects.
❑ The sequence in which the messages are flowing.
❑ Object organization.

The Sequence Diagram



The sequence diagram has four objects (Customer, Order, SpecialOrder and NormalOrder). This diagram shows the message sequence for *SpecialOrder* object and the same can be used in case of *NormalOrder* object.

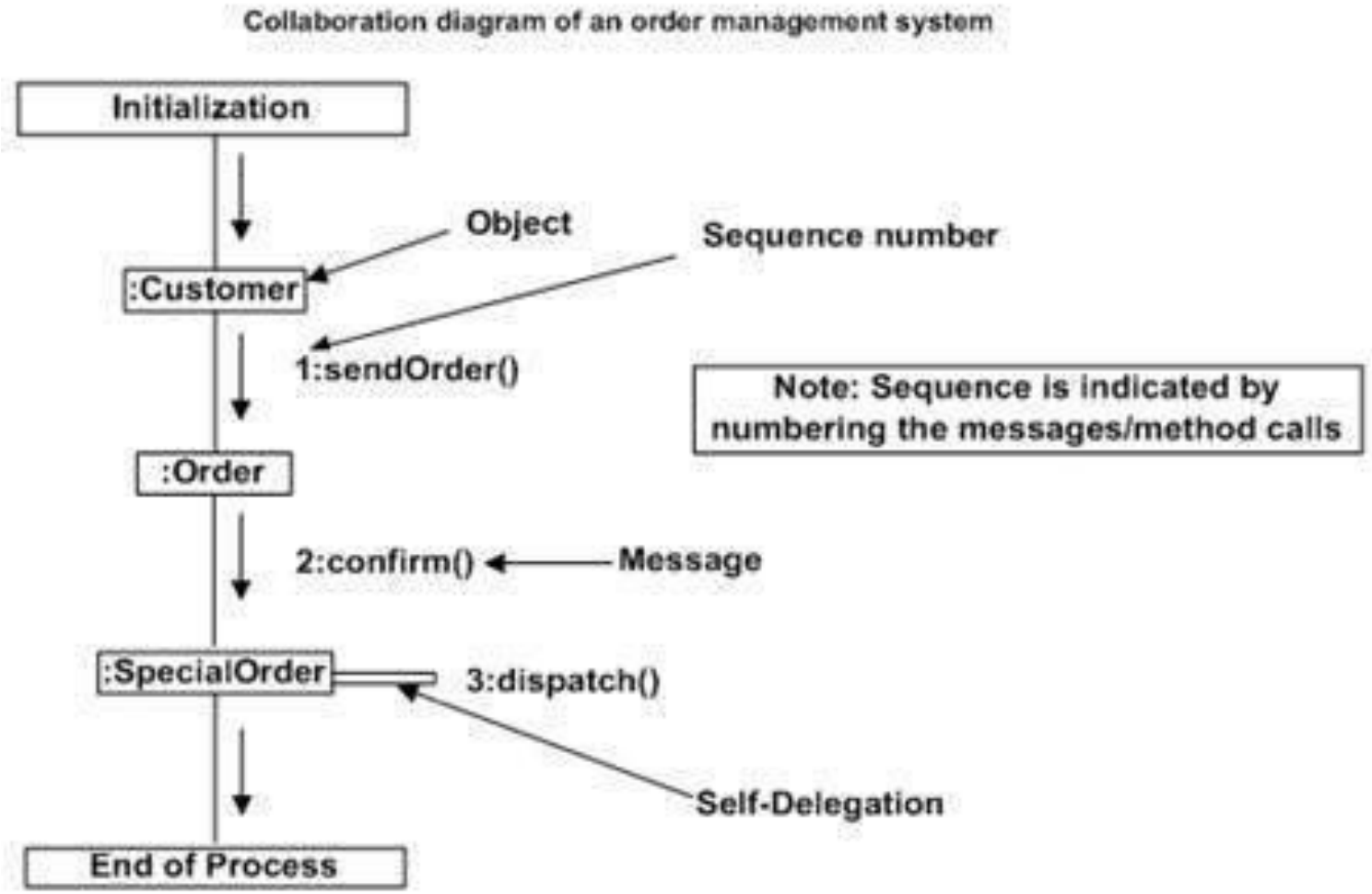
It is important to understand the time sequence of message flows. The message flow is nothing but a method call of an object.

The first call is *sendOrder ()* which is a method of *Order object*. The next call is *confirm ()* which is a method of *SpecialOrder* object and the last call is *Dispatch ()* which is a method of *SpecialOrder* object. The diagram mainly describes the method calls from one object to another, and this is also the actual scenario when the system is running.

The Collaboration Diagram

- The second interaction diagram is the collaboration diagram. It shows the object organization as seen in the following diagram. In the collaboration diagram, the method call sequence is indicated by some numbering technique. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram.
- Method calls are similar to that of a sequence diagram. However, difference being the sequence diagram does not describe the object organization, whereas the collaboration diagram shows the object organization.
- To choose between these two diagrams, emphasis is placed on the type of requirement. If the time sequence is important, then the sequence diagram is used. If organization is required, then collaboration diagram is used.

Sample Collaboration Diagram



Where to Use Interaction Diagrams?

- To understand the practical application, we need to understand the basic nature of sequence and collaboration diagram.
- The main purpose of both the diagrams are similar as they are used to capture the dynamic behavior of a system. However, the specific purpose is more important to clarify and understand.
- Sequence diagrams are used to capture the order of messages flowing from one object to another. Collaboration diagrams are used to describe the structural organization of the objects taking part in the interaction. A single diagram is not sufficient to describe the dynamic aspect of an entire system, so a set of diagrams are used to capture it as a whole.
- Interaction diagrams are used when we want to understand the message flow and the structural organization. Message flow means the sequence of control flow from one object to another. Structural organization means the visual organization of the elements in a system.

  Interaction diagrams can be used −

- ❑ To model the flow of control by time sequence.
- ❑ To model the flow of control by structural organizations.
- ❑ For forward engineering.
- ❑ For reverse engineering.

Statechart Diagrams

- The name of the diagram itself clarifies the purpose of the diagram and other details. It describes different states of a component in a system. The states are specific to a component/object of a system.
- A Statechart diagram describes a state machine. State machine can be defined as a machine which defines different states of an object and these states are controlled by external or internal events.

Purpose of Statechart Diagrams

- Statechart diagram is one of the five UML diagrams used to model the dynamic nature of a system. They define different states of an object during its lifetime and these states are changed by events.
- Statechart diagrams are useful to model the reactive systems. Reactive systems can be defined as a system that responds to external or internal events.
- Statechart diagram describes the flow of control from one state to another state. States are defined as a condition in which an object exists and it changes when some event is triggered.
- The most important purpose of Statechart diagram is to model lifetime of an object from creation to termination.
- Statechart diagrams are also used for forward and reverse engineering of a system. However, the main purpose is to model the reactive system.

Following are the main purposes of using Statechart diagrams −

- ❑ To model the dynamic aspect of a system.
- ❑ To model the life time of a reactive system.
- ❑ To describe different states of an object during its life time.
- ❑ Define a state machine to model the states of an object.

How to Draw a Statechart Diagram?

- Statechart diagram is used to describe the states of different objects in its life cycle. Emphasis is placed on the state changes upon some internal or external events. These states of objects are important to analyze and implement them accurately.
- Statechart diagrams are very important for describing the states. States can be identified as the condition of objects when a particular event occurs.
- Before drawing a Statechart diagram we should clarify the following points –

❑ Identify the important objects to be analyzed.

❑ Identify the states.

❑ Identify the events.

Sample Statechart Diagram

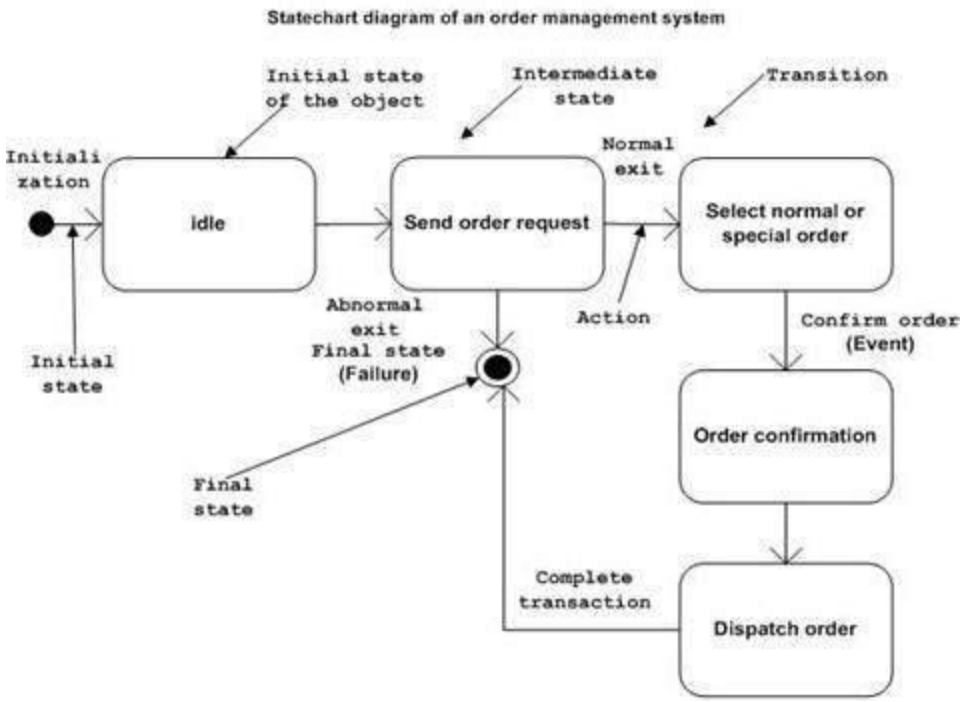
The first state is an idle state from where the process starts.

The next states are arrived for events like send request, confirm request, and dispatch order. These events are responsible for the state changes of order object.

During the life cycle of an object (here order object) it goes through the following states and there may be some abnormal exits.

This abnormal exit may occur due to some problem in the system. When the entire life cycle is complete, it is considered as a complete transaction as shown in the figure.

The initial and final state of an object is also shown in the figure.



Where to Use Statechart Diagrams?

- Statechart diagrams are used to model the dynamic aspect of a system like other four diagrams discussed in this tutorial. However, it has some distinguishing characteristics for modeling the dynamic nature.
- Statechart diagram defines the states of a component and these state changes are dynamic in nature. Its specific purpose is to define the state changes triggered by events. Events are internal or external factors influencing the system.
- Statechart diagrams are used to model the states and also the events operating on the system. When implementing a system, it is very important to clarify different states of an object during its life time and Statechart diagrams are used for this purpose. When these states and events are identified, they are used to model it and these models are used during the implementation of the system.
- If we look into the practical implementation of Statechart diagram, then it is mainly used to analyze the object states influenced by events. This analysis is helpful to understand the system behavior during its execution.

The main usage can be described as −

- ❑ To model the object states of a system.
- ❑ To model the reactive system. Reactive system consists of reactive objects.
- ❑ To identify the events responsible for state changes.
- ❑ Forward and reverse engineering.

UML - Activity Diagrams

- Activity diagram is another important diagram in UML to describe the dynamic aspects of the system.
- Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.
- The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.

Purpose of Activity Diagrams

- The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.
- Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.
- It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

- ❑ Draw the activity flow of a system.
- ❑ Describe the sequence from one activity to another.
- ❑ Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

- Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.
- Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

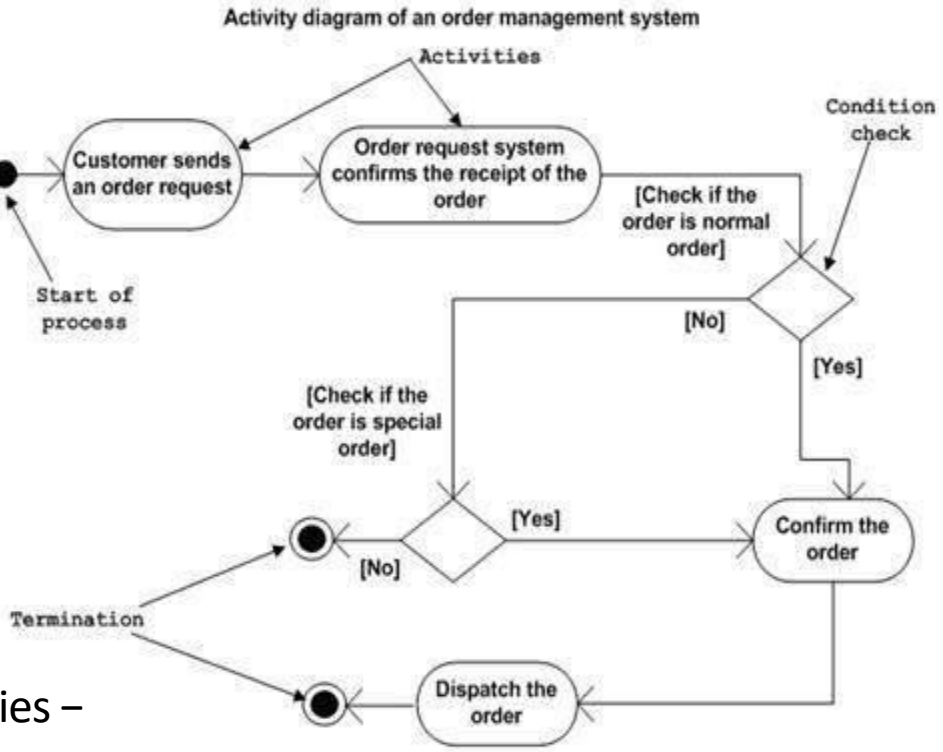
  Before drawing an activity diagram, we should identify the following elements −

- ❑ Activities
- ❑ Association
- ❑ Conditions
- ❑ Constraints

- Once the above-mentioned parameters are identified, we need to make a mental layout of the entire flow. This mental layout is then transformed into an activity diagram.

Sample Activity Diagram

Here is an example of an activity diagram for order management system. In the diagram, four activities are identified which are associated with conditions. One important point should be clearly understood that an activity diagram cannot be exactly matched with the code. The activity diagram is made to understand the flow of activities and is mainly used by the business users



This diagram is drawn with the four main activities –

- Send order by the customer
- Receipt of the order
- Confirm the order
- Dispatch the order

After receiving the order request, condition checks are performed to check if it is normal or special order. After the type of order is identified, dispatch activity is performed and that is marked as the termination of the process.

Where to Use Activity Diagrams?

- The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.
- Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.
- Now, it is clear that an activity diagram is drawn from a very high level. So it gives high level view of a system. This high level view is mainly for business users or any other person who is not a technical person.
- This diagram is used to model the activities which are nothing but business requirements. The diagram has more impact on business understanding rather than on implementation details.

Activity diagram can be used for −

- ❑ Modeling work flow by using activities.
- ❑ Modeling business requirements.
- ❑ High level understanding of the system's functionalities.
- ❑ Investigating business requirements at a later stage.