Description of Different Diagrams

Dr. Buddhadeb Pradhan

Class Diagram

- Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application.
- Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.
- Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

Purpose of Class Diagrams

- The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.
- UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

✓ The purpose of the class diagram can be summarized as –

✓ Analysis and design of the static view of an application.

✓ Describe responsibilities of a system.

✓ Base for component and deployment diagrams.

✓ Forward and reverse engineering.

How to Draw a Class Diagram?

- Class diagrams are the most popular UML diagrams used for construction of software applications. It is very important to learn the drawing procedure of class diagram.
- Class diagrams have a lot of properties to consider while drawing but here the diagram will be considered from a top level view.
- Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. A collection of class diagrams represent the whole system.

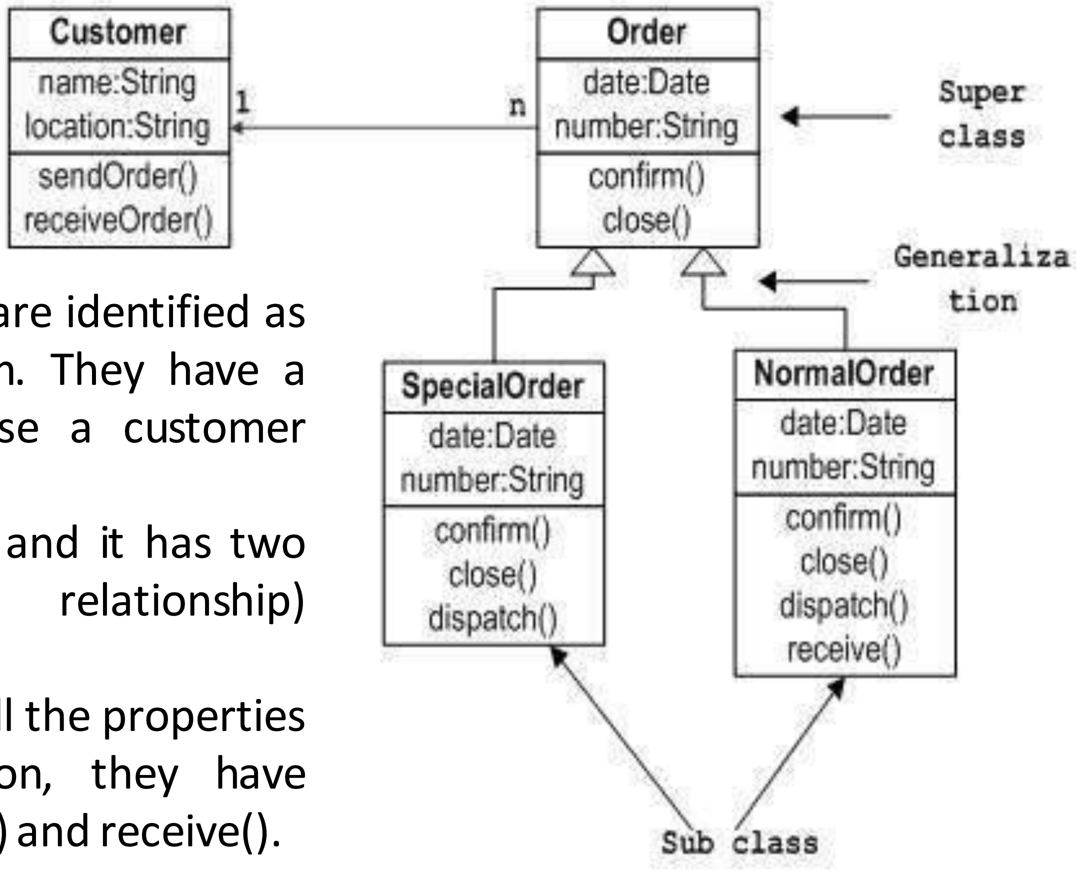
How to Draw a Class Diagram?

The following points should be remembered while drawing a class diagram −

- The name of the class diagram should be meaningful to describe the aspect of the system.
- Each element and their relationships should be identified in advance.
- Responsibility (attributes and methods) of each class should be clearly identified
- For each class, minimum number of properties should be specified, as unnecessary properties will make the diagram complicated.
- Use notes whenever required to describe some aspect of the diagram. At the end of the drawing it should be understandable to the developer/coder.
- Finally, before making the final version, the diagram should be drawn on plain paper and reworked as many times as possible to make it correct.

Class Diagram: Order System of an application

Sample Class Diagram



- First of all, Order and Customer are identified as the two elements of the system. They have a one-to-many relationship because a customer can have multiple orders.
- Order class is an abstract class and it has two concrete classes (inheritance relationship) SpecialOrder and NormalOrder.
- The two inherited classes have all the properties as the Order class. In addition, they have additional functions like dispatch() and receive().

Where to Use Class Diagrams?

- Class diagram is a static diagram and it is used to model the static view of a system. The static view describes the vocabulary of the system.
- Class diagram is also considered as the foundation for component and deployment diagrams. Class diagrams are not only used to visualize the static view of the system but they are also used to construct the executable code for forward and reverse engineering of any system.
- Generally, UML diagrams are not directly mapped with any object-oriented programming languages but the class diagram is an exception.
- Class diagram clearly shows the mapping with object-oriented languages such as Java, C++, etc. From practical experience, class diagram is generally used for construction purpose.

❑ In a nutshell it can be said, class diagrams are used for −

❑ Describing the static view of the system.

❑ Showing the collaboration among the elements of the static view.

❑ Describing the functionalities performed by the system.

❑ Construction of software applications using object oriented languages.

Object Diagrams

- Object diagrams are derived from class diagrams so object diagrams are dependent upon class diagrams.
- Object diagrams represent an instance of a class diagram. The basic concepts are similar for class diagrams and object diagrams.
- Object diagrams also represent the static view of a system but this static view is a snapshot of the system at a particular moment.
- Object diagrams are used to render a set of objects and their relationships as an instance.

Purpose of Object Diagrams

- The purpose of a diagram should be understood clearly to implement it practically. The purposes of object diagrams are similar to class diagrams.
- The difference is that a class diagram represents an abstract model consisting of classes and their relationships. However, an object diagram represents an instance at a particular moment, which is concrete in nature.
- It means the object diagram is closer to the actual system behavior. The purpose is to capture the static view of a system at a particular moment.

✓ The purpose of the object diagram can be summarized as −

✓ Forward and reverse engineering.

✓ Object relationships of a system

✓ Static view of an interaction.

✓ Understand object behaviour and their relationship from practical perspective

How to Draw an Object Diagram?

- To capture a particular system, numbers of class diagrams are limited. However, if we consider object diagrams then we can have unlimited number of instances, which are unique in nature.
- Only those instances are considered, which have an impact on the system.
- A single object diagram cannot capture all the necessary instances or rather cannot specify all the objects of a system.

How to Draw an Object Diagram?

- First, analyze the system and decide which instances have important data and association.
- Second, consider only those instances, which will cover the functionality.
- Third, make some optimization as the number of instances are unlimited.

How to Draw an Object Diagram?

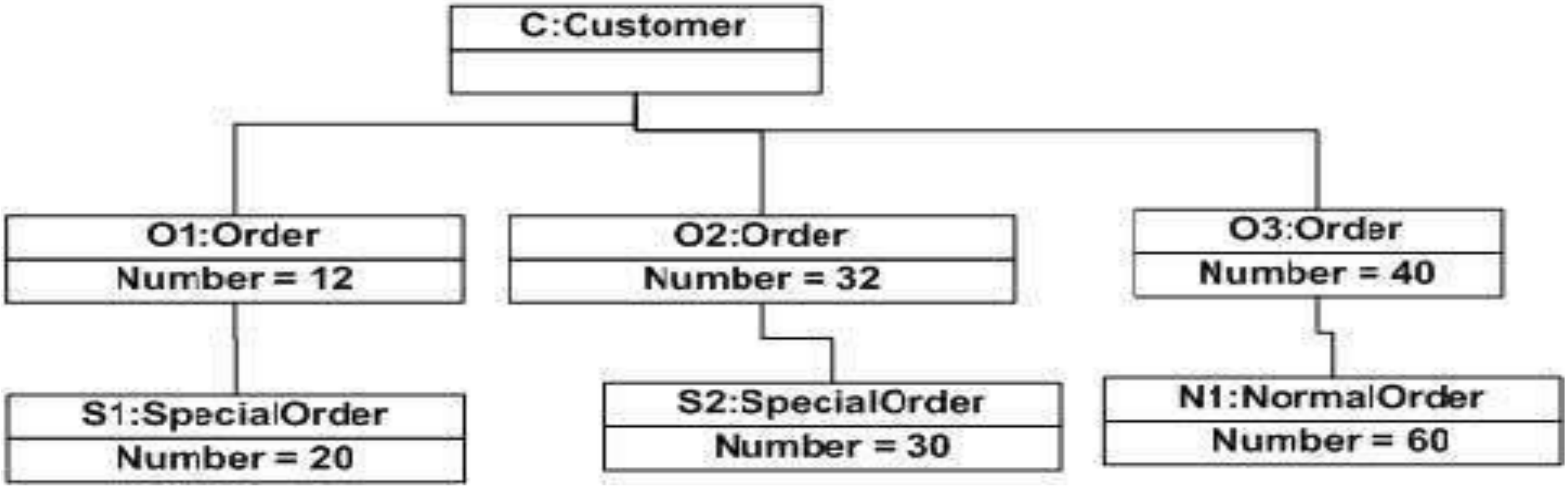
- Before drawing an object diagram, the following things should be remembered and understood clearly –
- Object diagrams consist of objects.
- The link in object diagram is used to connect objects.
- Objects and links are the two elements used to construct an object diagram.

How to Draw an Object Diagram?

The following things are to be decided before starting the construction of the diagram –

- The object diagram should have a meaningful name to indicate its purpose.
- The most important elements are to be identified.
- The association among objects should be clarified.
- Values of different elements need to be captured to include in the object diagram.
- Add proper notes at points where more clarity is required.

Object diagram of an order management system



- Now the customer object (C) is associated with three order objects (O1, O2, and O3). These order objects are associated with special order and normal order objects (S1, S2, and N1). The customer has the following three orders with different numbers (12, 32 and 40) for the particular time considered.
- The customer can increase the number of orders in future and in that scenario the object diagram will reflect that. If order, special order, and normal order objects are observed then you will find that they have some values.
- For orders, the values are 12, 32, and 40 which implies that the objects have these values for a particular moment (here the particular time when the purchase is made is considered as the moment) when the instance is captured
- The same is true for special order and normal order objects which have number of orders as 20, 30, and 60. If a different time of purchase is considered, then these values will change accordingly.

Where to Use Object Diagrams?

- Object diagrams can be imagined as the snapshot of a running system at a particular moment. Let us consider an example of a running train
- Now, if you take a snap of the running train then you will find a static picture of it having the following −

❑ A particular state which is running.

❑ A particular number of passengers. which will change if the snap is taken in a different time

- Here, we can imagine the snap of the running train is an object having the above values. And this is true for any real-life simple or complex system.
- In a nutshell, it can be said that object diagrams are used for −

❖ Making the prototype of a system.

❖ Reverse engineering.

❖ Modeling complex data structures.

❖ Understanding the system from practical perspective.

Component Diagrams

- Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.
- Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose of Component Diagrams

- Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities.
- Thus from that point of view, component diagrams are used to visualize the physical components in a system. These components are libraries, packages, files, etc.
- Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organization of the components at a particular moment.
- A single component diagram cannot represent the entire system but a collection of diagrams is used to represent the whole.

❑ The purpose of the component diagram can be summarized as −

❑ Visualize the components of a system.

❑ Construct executables by using forward and reverse engineering.

❑ Describe the organization and relationships of the components.

How to Draw a Component Diagram?

- Component diagrams are used to describe the physical artifacts of a system. This artifact includes files, executables, libraries, etc
- The purpose of this diagram is different. Component diagrams are used during the implementation phase of an application. However, it is prepared well in advance to visualize the implementation details.
- Initially, the system is designed using different UML diagrams and then when the artifacts are ready, component diagrams are used to get an idea of the implementation.
- This diagram is very important as without it the application cannot be implemented efficiently. A well-prepared component diagram is also important for other aspects such as application performance, maintenance, etc.

How to Draw a Component Diagram?

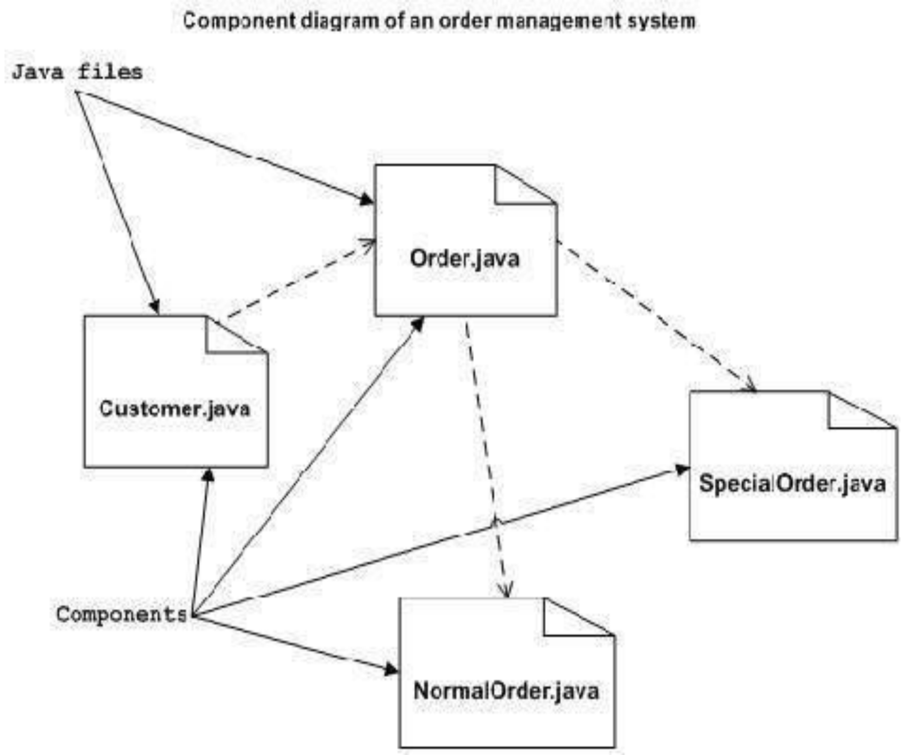
Before drawing a component diagram, the following artifacts are to be identified clearly −

- ❑ Files used in the system.
- ❑ Libraries and other artifacts relevant to the application.
- ❑ Relationships among the artifacts.

After identifying the artifacts, the following points need to be kept in mind.

- ✓ Use a meaningful name to identify the component for which the diagram is to be drawn.
- ✓ Prepare a mental layout before producing the using tools.
- ✓ Use notes for clarifying important points.

Sample Component Diagram



Following is a component diagram for order management system. Here, the artifacts are files. The diagram shows the files in the application and their relationships. In actual, the component diagram also contains dlls, libraries, folders, etc.

In the following diagram, four files are identified and their relationships are produced. Component diagram cannot be matched directly with other UML diagrams discussed so far as it is drawn for completely different purpose.

Where to Use Component Diagrams?

- We have already described that component diagrams are used to visualize the static implementation view of a system. Component diagrams are special type of UML diagrams used for different purposes.
- These diagrams show the physical components of a system. To clarify it, we can say that component diagrams describe the organization of the components in a system.
- Organization can be further described as the location of the components in a system. These components are organized in a special way to meet the system requirements.
- As we have already discussed, those components are libraries, files, executables, etc. Before implementing the application, these components are to be organized. This component organization is also designed separately as a part of project execution.
- Component diagrams are very important from implementation perspective. Thus, the implementation team of an application should have a proper knowledge of the component details
- Component diagrams can be used to −

✓ Model the components of a system.

✓ Model the database schema.

✓ Model the executables of an application.

✓ Model the system's source code.