## Architectural Modeling

Architectural model represents the overall framework of the system. It contains both structural and behavioral elements of the system. Architectural model can be defined as the blueprint of the entire system. Package diagram comes under architectural modeling.

**Component Diagrams**

## 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.

## 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.

# UML - Deployment Diagrams

## 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.

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.

### Collaboration Diagram

* Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.
* The purpose of collaboration diagram is similar to sequence diagram. However, the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.

## UML patterns and framework

* The generalized patterns framework is the default framework that defines the common default behavior across all pattern implementations. This framework implements the necessary service protocols that activate the pattern in the product. The core pattern plug-in components contain the default framework that provides the base classes that pattern implementations specialize.
* The pattern structure is defined in terms of both UML and Reusable Asset Specification (RAS) asset metamodels. A UML representation of the pattern persists in the pattern plug-in project. The model-operative patterns operate on UML models and their elements.
* The pattern service discovers the available pattern plug-ins from several sources, including installed plug-ins and local or remote RAS repositories. The pattern service also identifies pattern definitions, creates pattern instances, and directly supports the client UI components. Both the pattern service and the pattern framework are Eclipse plug-ins.

## Systems and Models in UML

**System** − A set of elements organized to achieve certain objectives form a system. Systems are often divided into subsystems and described by a set of models.

**Model** − Model is a simplified, complete, and consistent abstraction of a system, created for better understanding of the system.

**View** − A view is a projection of a system’s model from a specific perspective.