

Types of modeling diagrams

You can use modeling diagrams to represent different views of the system, application, or database that you are developing. Several different diagrams can illustrate multiple views of a model.

The following table describes the different types of modeling diagrams that you can use to illustrate models:

Diagram	Description
Activity	An activity diagram is a UML behavioral diagram that shows the steps or tasks that constitute a process. An activity diagram is similar to a flowchart and provides a view of the flow between activities in a process. Activity diagrams model the dynamic parts of a system and show how activities interact with each other.
Browse	A browse diagram is a temporary, non-editable, non-UML diagram that the product automatically creates from a query. You can use a browse diagram to navigate through a model to view the details of its model elements. For example, you can browse a specific class in your diagram and see the relationships that it has with other model elements. You can then select one of these model elements and open a browse diagram to view its relationships.
Class	A class diagram is a UML structural diagram. Depending on the complexity of a system, you can use a single class diagram to model the entire system, or you can use several class diagrams to model the components of the system. Class diagrams are the blueprints of your system. Use class diagrams to model the objects that make up the system, to display the relationships between the objects, and to describe what those objects do.
Communication	A communication diagram is a UML behavioral diagram that shows the interaction between the objects or roles that are associated with a lifeline and shows the passing of messages. In earlier versions of UML, this diagram was called a collaboration diagram and had a different notation.
Component	A component diagram is a UML structural diagram that provides a view of the physical software components in a software system, their interfaces, and their dependencies. Components are physical, replaceable parts of a system, such as executable files, libraries, software components (for example, ActiveX or JavaBeans components), documents, databases, and files. A component diagram can depict all or some of the components in a system.
Composite structure	A composite structure diagram is a UML structural diagram that defines the parts of a system and the communication relationships between them.
Deployment	A deployment diagram is a UML structural diagram that models the physical nodes where your system executes or supports execution, and the physical connections and relationships of the nodes. Physical nodes can include common devices such as servers and workstations, as well as other devices that support the runtime environment of a system. Deployment diagrams document the physical relationships between the software and hardware components in the system, and the physical distribution of the processing.
Freeform	A freeform diagram is a non-UML diagram that can contain a combination of UML and non-UML diagram elements such as classes, components, and geometric shapes. You can use freeform diagrams to show any part of your software system early in the software development process. For example, in a freeform diagram, you can use the geometric shapes such as ovals, rectangles, and triangles to create an organizational chart for your project team. You can create some parts of your model or diagram with UML shapes and create other parts with shapes that address your specific needs. You can later evolve the diagrams to be compliant with UML. Note: You cannot add model elements that belong to UML state machines, activities, and interactions to freeform diagrams.
Interaction overview	An interaction overview diagram is a UML behavioral diagram that defines interactions and is a variant of the activity diagram, which emphasizes the high-level control flow. Interaction overview diagrams illustrate an overview of a flow of control in which each node can be an interaction diagram.
Object	An object diagram is a UML structural diagram that is similar to a class diagram except that, instead of dealing with classes, it shows objects that are instances of classes.
Package	A package diagram is a UML structural diagram that shows how model elements are organized into packages and that shows the dependencies between them, including package imports and package merges.
Sequence	A sequence diagram is a UML structural diagram that provides a view of the chronological sequence of messages between objects or classifier roles that work together in an interaction or interaction instance. A sequence diagram consists of a group of instances, which are represented by lifelines, and the messages that they exchange during the interaction.
State machine	A state machine diagram, which was called a state chart diagram in earlier versions of UML, is a UML behavioral diagram. A state machine diagram is a graphical representation of the sequence of states of an object, the events that cause a transition from one state to another, and the actions that result from a change in state. You can add diagrams to a state machine to describe different behavioral aspects of an object.
Timing	A timing diagram is a UML behavioral diagram that shows the change in state or other condition of a lifeline, which represents a classifier instance or classifier role, over time.
Topic	A topic diagram is a non-editable, non-UML diagram that provides a quick way to show existing relationships between elements. You can use a topic diagram to define a query and to display and persist the results from that query. For example, you can create a hierarchy topic diagram that you can then update to always show the hierarchy for a specific class. You can also use this diagram to show traceability information. A topic diagram is similar to a browse diagram except that you can save the topic diagram in the model and then reopen it. Any changes that you make to the model elements are reflected in the refreshed topic diagram.
Use case	A use case diagram is a UML behavioral diagram that focuses on the requirements and describes the high-level functions and scope of a system. These diagrams identify the users and show interactions between the system and the user. Use case diagrams can depict an entire system or only significant portions of the system. The use cases and actors in use case diagrams describe how a user uses a system, not how the system operates internally.

Related concepts

[Modeling diagrams](#)

Feedback

(C) Copyright IBM Corporation 2004, 2005. All Rights Reserved.