UML Diagram

UML stands for Unified Modeling Language. UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady Booch, James Rumbaugh, Ivar Jacobson, and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. Today, UML is accepted by the [Object Management Group (OMG)](http://www.omg.org/) as the standard for modeling software development.

What's new in UML 2.0

UML 2.0 helped extend the original UML specification to cover a wider portion of software development efforts including agile practices.

Here are some of the changes made to [UML diagrams](https://www.smartdraw.com/uml-diagram/) in UML 2.0:

* Improved integration between structural models like class diagrams and behavior models like activity diagrams.
* Added the ability to define a hierarchy and decompose a software system into components and sub-components.
* The original UML specified nine diagrams; UML 2.x brings that number up to 13. The four new diagrams are called: communication diagram, composite structure diagram, interaction overview diagram, and timing diagram. It also renamed statechart diagrams to state machine diagrams, also known as state diagrams.

Types of [UML Diagrams](https://www.smartdraw.com/uml-diagram/)

The current UML standards call for 13 different types of diagrams: class, activity, object, use case, sequence, package, state, component, communication, composite structure, interaction overview, timing, and deployment.

These diagrams are organized into two distinct groups: structural diagrams and behavioral or interaction diagrams.

Structural [UML diagrams](https://www.smartdraw.com/uml-diagram/)

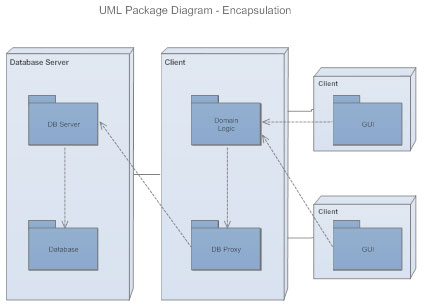
* Class diagram
* Package diagram
* Object diagram
* Component diagram
* Composite structure diagram
* Deployment diagram

Behavioral [UML diagrams](https://www.smartdraw.com/uml-diagram/)

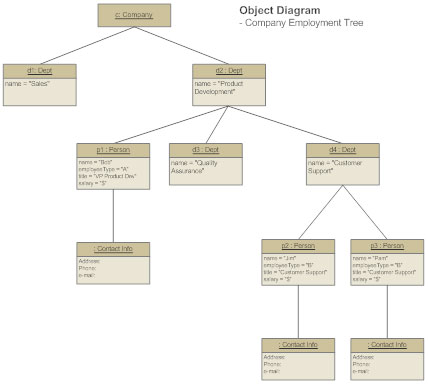
* Activity diagram
* Sequence diagram
* Use case diagram
* State diagram
* Communication diagram
* Interaction overview diagram
* Timing diagram

Class Diagram   
[Class diagrams](https://www.smartdraw.com/class-diagram/) are the backbone of almost every object-oriented method, including UML. They describe the static structure of a system. [Learn more](https://www.smartdraw.com/class-diagram/)

Package Diagram  
Package diagrams are a subset of class diagrams, but developers sometimes treat them as a separate technique. Package diagrams organize elements of a system into related groups to minimize dependencies between packages.

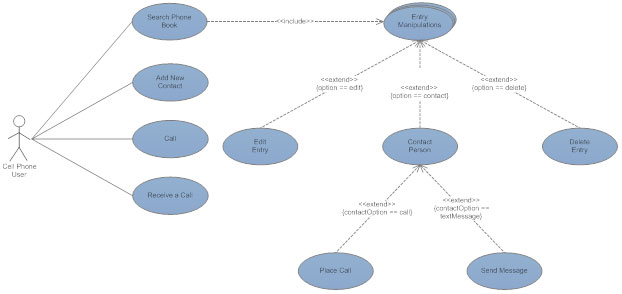


Object Diagram  
Object diagrams describe the static structure of a system at a particular time. They can be used to test class diagrams for accuracy.

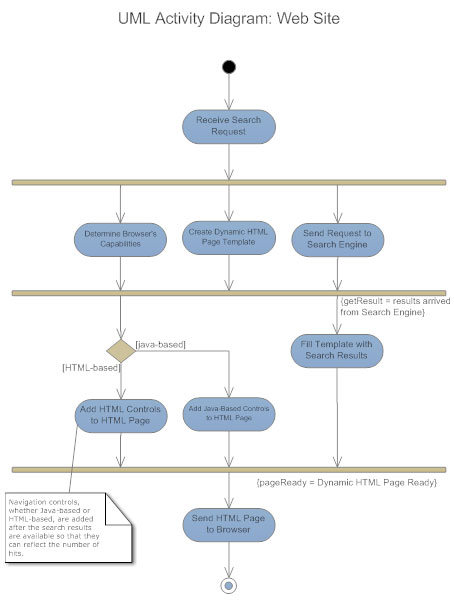


Composite Structure Diagram  
Composite structure diagrams show the internal part of a class.

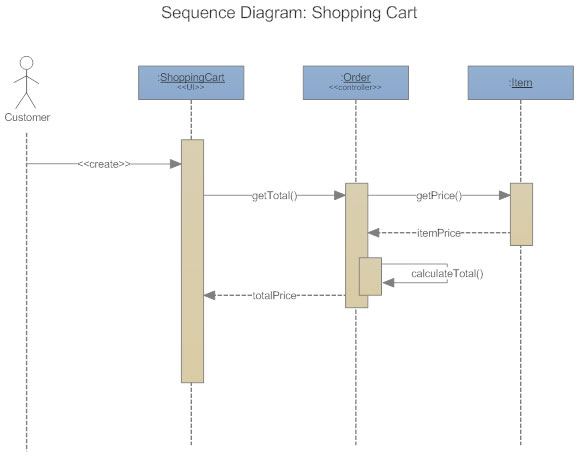
Use Case Diagram   
[Use case diagrams](https://www.smartdraw.com/use-case-diagram/) model the functionality of a system using actors and use cases. [Learn more](https://www.smartdraw.com/use-case-diagram/)



Activity Diagram  
[Activity diagrams](https://www.smartdraw.com/activity-diagram/) illustrate the dynamic nature of a system by modeling the flow of control from activity to activity. An activity represents an operation on some class in the system that results in a change in the state of the system. Typically, activity diagrams are used to model workflow or business processes and internal operation. [Learn more](https://www.smartdraw.com/activity-diagram/)



Sequence Diagram  
[Sequence diagrams](https://www.smartdraw.com/sequence-diagram/) describe interactions among classes in terms of an exchange of messages over time. [Learn more](https://www.smartdraw.com/sequence-diagram/)

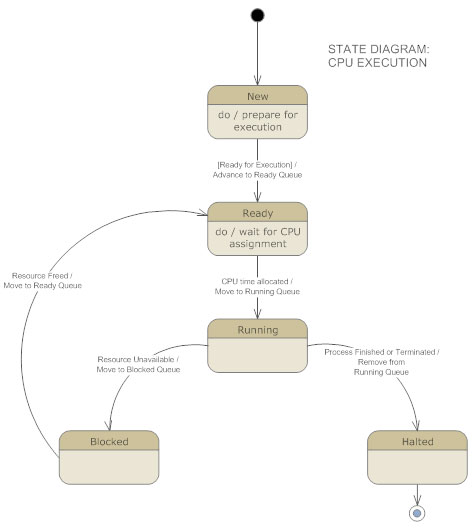


Interaction Overview Diagram  
Interaction overview diagrams are a combination of activity and sequence diagrams. They model a sequence of actions and let you deconstruct more complex interactions into manageable occurrences. You should use the same notation on interaction overview diagrams that you would see on an activity diagram.

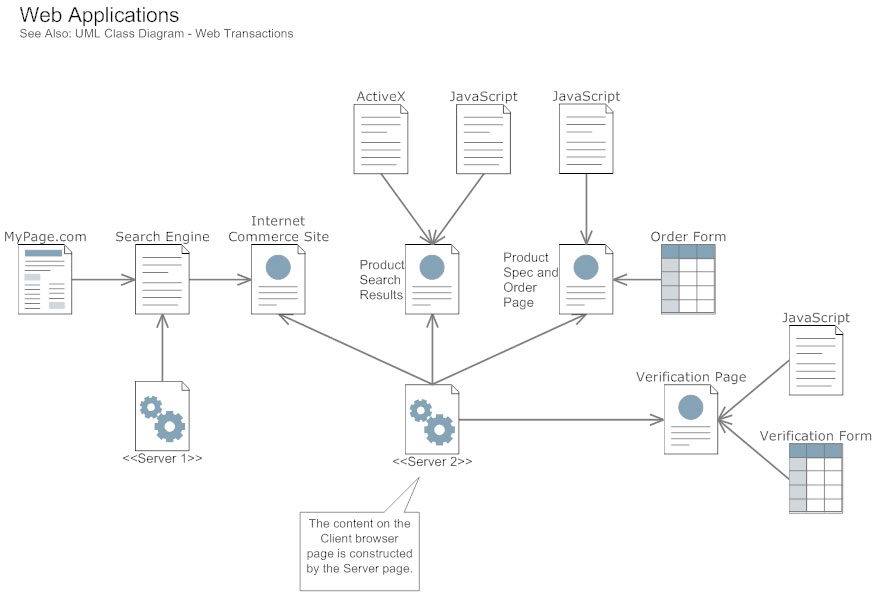
Timing Diagram  
A timing diagram is a type of behavioral or interaction UML diagram that focuses on processes that take place during a specific period of time. They're a special instance of a sequence diagram, except time is shown to increase from left to right instead of top down.

Communication Diagram  
Communication diagrams model the interactions between objects in sequence. They describe both the static structure and the dynamic behavior of a system. In many ways, a communication diagram is a simplified version of a collaboration diagram introduced in UML 2.0.

State Diagram  
[Statechart diagrams](https://www.smartdraw.com/state-diagram/), now known as state machine diagrams and state diagrams describe the dynamic behavior of a system in response to external stimuli. State diagrams are especially useful in modeling reactive objects whose states are triggered by specific events. [Learn more](https://www.smartdraw.com/state-diagram/)



Component Diagram  
[Component diagrams](https://www.smartdraw.com/component-diagram/) describe the organization of physical software components, including source code, run-time (binary) code, and executables.[Learn more](https://www.smartdraw.com/component-diagram/).



Deployment Diagram  
Deployment diagrams depict the physical resources in a system, including nodes, components, and connections.

Why You Should Diagram Before You Code

A complex enterprise application with many collaborators will require a solid foundation of planning and clear, concise communication among team members as the project progresses.

Visualizing user interactions, processes, and the structure of the system you're trying to build will help save time down the line and make sure everyone on the team is on the same page.