我们将 UML 中的图分为两大类：

结构图

行为图

## **UML 结构图：**

UML 结构图表示系统的静态方面。这些静态方面指示，形成的主要结构并因此稳定那些部分。

这些静态部分是类，接口，对象，组件和节点四个结构图：

类图

对象图

组件图

部署图

### **UML 类图：**

类图是描述系统中的类，以及各个类之间的关系的静态视图，它包括：类、接口、关联和协作。

类图能够让我们在正确编写代码以前对系统有一个全面的认识。

类图是一种模型类型，确切的说，是一种静态模型类型。

活动类在类图来表示系统的并发性。

类图代表的面向对象的系统。

### **UML 对象图：**

对象图与类图极为相似，它是类图的实例，对象图显示类的多个对象实例，而不是实际的类，它描述的不是类之间的关系，而是对象之间的关系。从实际的角度来看，它们被用来建立一个系统的原型。

### **UML 组件图：**

组件图描述代码构件的物理结构以及各种构建之间的依赖关系。

组件图用来建模软件的组件及其相互之间的关系，这些图由构件标记符和构件之间的关系构成。

组件图中，构件时软件单个组成部分，它可以是一个文件，产品、可执行文件和脚本等。

在设计阶段的软件构件（类，接口等）的系统被安排在不同的组，这取决于他们的关系。这些组被称为组件。

组件图用于可视化的实现。

### **UML 部署图：**

部署图是用来建模系统的物理部署。例如计算机和设备，以及它们之间是如何连接的。

部署图的使用者是开发人员、系统集成人员和测试人员。

部署图是一组节点和它们之间的关系，这些节点部署这些组件的物理实体。

部署图用于可视化系统的部署视图。

***注:***如果上述描述和用法仔细观察，这是很清楚的，所有的图表都彼此有某种关系。组件图是依赖的类，接口等类/对象图的一部分。再次部署图是取决于使用的组件，这些组件，以使一个组件图。

## **UML 行为图：**

任何系统都可以有两个方面，静态和动态。因此，一个模型被认为是完成时，这两个方面都完全覆盖。

行为图基本上捕捉系统的动态方面。动态方面可以进一步改变/移动系统的一部分。

UML具有以下五种行为图：

用例图

序列图

协作图

状态图

活动图

### **UML 用例图：**

用例图描述角色以及角色与用例之间的连接关系。说明的是谁要使用系统，以及他们使用该系统可以做些什么。

一个用例图包含了多个模型元素，如系统、参与者和用例，并且显示了这些元素之间的各种关系，如泛化、关联和依赖。

因此，用例图是用来描述的功能之间的关系和他们的内部/外部控制器，这些控制器是已知的参与者。

### **UML 序列图：**

序列图是一种交互图。

序列图是用来显示你的参与者如何以一系列顺序的步骤与系统的对象交互的模型。

序列图可以用来展示对象之间是如何进行交互的。

序列图将显示的重点放在消息序列上，即强调消息是如何在对象之间被发送和接收的。

从实施和执行的角度来看是非常重要的系统组件之间的交互。

因此，在一个系统中执行一个特定的功能的调用序列的序列图是用于可视化。

### **UML 协作图：**

协作图和序列图相似，是另一种形式的交互图；如果强调时间和顺序，则使用序列图；如果强调上下级关系，则选择协作图。

协作图代表了一个系统的组织结构和发送/接收的消息。组织结构由对象和链接。

协作图的目的是类似的序列图。但是，协作图的具体目的是可视化的组织对象及其相互作用。

### **UML 状态图：**

状态图描述类的对象所有可能的状态，以及事件发生时状态的转移条件。

状态图可以捕获对象、子系统和系统的生命周期。他们可以告知一个对象可以拥有的状态，并且事件(如消息的接收、时间的流逝、错误、条件变为真等)会怎么随着时间的推移来影响这些状态。

状态图是用来表示的事件驱动的系统状态的变化。它基本上描述了类，接口状态变化等

状态图是用于可视化的反应系统内部/外部因素。

### **UML 活动图：**

活动图描述了在一个系统中的控制流。

活动图描述用例要求所要进行的活动，以及活动间的约束关系，有利于识别并行活动。

活动图能够演示出系统中哪些地方存在功能，以及这些功能和系统中其他组件的功能如何共同满足前面使用用例图建模的商务需求。

活动图用于可视化的流量控制在一个系统中。这是准备系统将如何工作，在执行时有一个想法。

***注:***在一个系统中很难捕捉到动态性质，而 UML 已经提供从不同的角度捕捉到动态系统的功能。序列图和协作图是同构的，它们之间的转换不会丢失任何信息。