

顺序图/时序图

原创xyzso1z最后发布于2019-03-05 19:47:37阅读数 1121☆ 收藏5编辑展开

1.定义

顺序图也称时序图，定义是：**顺序图是显示对象之间交互的图，这些对象是按时间顺序排列的。特别地，顺序图中显示的是参与交互的对象及对象之间消息交互的顺序。**

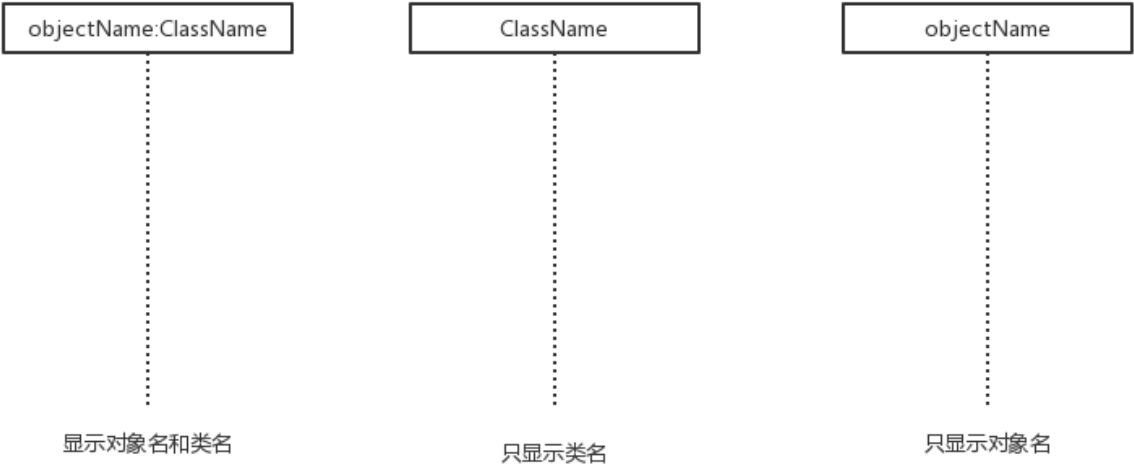
目的：**序列图的主要用途之一，是把用例表达的需求，转化为进一步、更加正式层次的精细表达。用例常常被细化为一个或者更多的序列图。序列图除了在设计新系统方面的用途外，它们还能用来记录一个存在系统（称它为“遗产”）的对象现在如何交互。当把这个系统移交给另一个人或组织时，这个文档很有用。**

顺序图是一个二维图形。在顺序图中水平方向为对象维，沿水平方向排列的是参与交互的对象。其中对象间的排列顺序并不重要，但一般把表示参与者的对象放在图的两侧，主要参与者放在最左边，次要参与者放在最右边（或表示人的参与者放在最左边，表示系统的参与者放在最右边）。顺序图中的垂直方向为时间维，沿垂直向下方向按时间递增顺序列出各对象所发出和接收的消息。

2.基本模块

1. 顺序图中包括的建模元素有：对象（参与者实例也是对象）、生命线（lifeline）、控制焦点（focus of control ,FOC）、消息等。

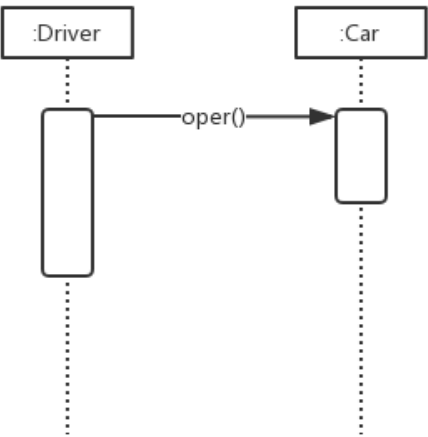
顺序图中对象的命名方式主要有3种，如图：



<https://blog.csdn.net/xyzso1z>

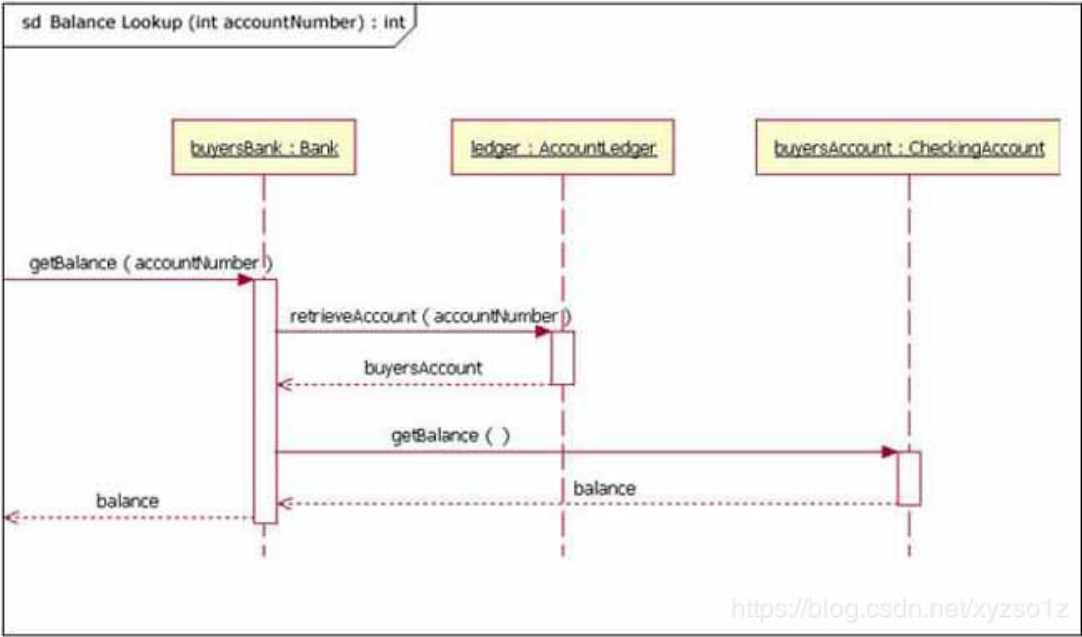
2. 生命线在顺序图中表示为从对象图标向下延伸的一条虚线，表示对象存在的时间，如上图。

3. 控制焦点是顺序图中表示时间段的符号，在这个时间段内，对象将执行对应的操作。控制焦点表示为生命线上的小矩形,如下图：



<https://blog.csdn.net/xyzso1z>

控制焦点可以嵌套，嵌套的控制焦点可以更精确地说明消息的开始和结束位置，如下图：



<https://blog.csdn.net/xyzso1z>

与焦点控制相关的概念是激活期（activation）。激活期表示对象执行一个动作的期间，即对象激活的时间段。根据定义可知，控制焦点和激活期事实上表示同一个意思。

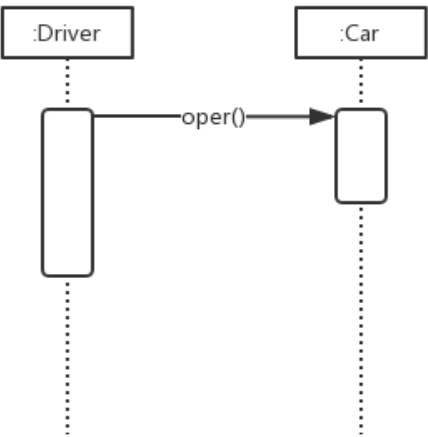
3.顺序图中的消息

顺序图中的重要概念是消息。分为三种**调用消息**、**异步消息**和**返回消息**这三种消息。

3.1 调用消息

调用消息（procedure call）发送者把控制传递给消息的接收者，然后停止活动，等待消息接收者放弃或返回控制。**调用消息可以用来表示同步的意义。**

调用消息的表示符号如下图，其中oper()是一个调用消息。

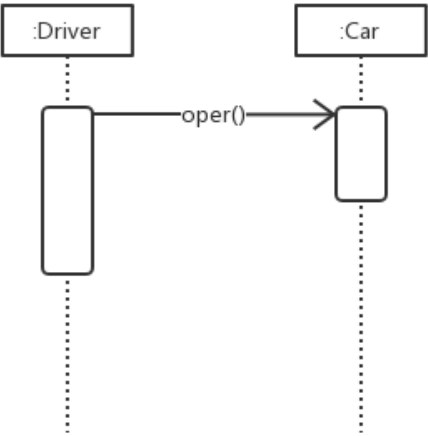


<https://blog.csdn.net/xyzso1z>

一般地，调用消息的接收者必须是一个被动对象（passive object），即它是一个需要通过消息驱动才能执行动作的对象。另外调用消息必有一个配对的返回消息，为了图的简洁和清晰，与调用消息配对的返回消息可以不用画出。

3.2 异步消息

异步消息（asynchronous）的发送者通过消息把信号传递给消息的接收者，然后继续自己的活动，不等待接收者返回消息或控制。异步消息的接收者和发送者时并发工作的。

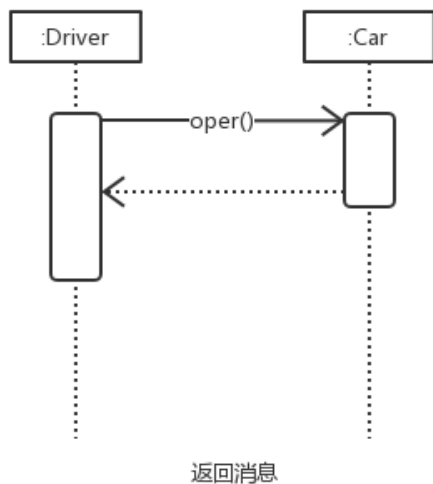


<https://blog.csdn.net/xyzso1z>

3.3 返回消息

返回消息（return）表示从过程调用返回。如果是从过程调用返回，则返回消息是隐含的，所以返回消息可以不用画出来。对于非过程调用，如果有返回消息，必须明确表示出来。如下图所示时返回消息的表示符号，其中的虚线

箭头表示对应于oper()这个消息的返回消息。



<https://blog.csdn.net/xyzso1z>

4.建立顺序图的步骤

在分析和设计过程中，建立顺序图并没有一个标准的步骤，下面给出的步骤只是指导性原则：

1. 确定交互过程的上下文（context）。
2. 识别参与交互过程的对象。
3. 为每个对象设置生命线，即确定哪些对象存在于整个交互过程中，哪些对象在交互过程中被创建和撤销。
4. 从引发这个交互过程的初始消息开始，在生命线之间自顶向下依次画出随后的各个消息。
5. 如果需要表示消息的嵌套，或 / 和表示消息发生时的时间点，则采用控制焦点。
6. 如果需要说明时间约束，则在消息旁边加上约束说明。
7. 如果需要，可以为每个消息附上前置条件和后置条件。