

UML 指南：顺序图

Robert C. Martin
Engineering Notebook
98.04

翻译：杨健

在上一期的专栏里，我描述了 UML 的协作图。协作图允许设计者确定在一个协作中对象间传递消息的顺序。这种图的风格相对于消息的顺序更注重对象间的关系。

在这一期里，我们将讨论 UML 的顺序图。顺序图装载了和协作图同样的信息，但注重的是消息的顺序，而不是对象间的关系。

回顾蜂窝电话

下面是上期的蜂窝电话例子的最终协作图。（见图 1）

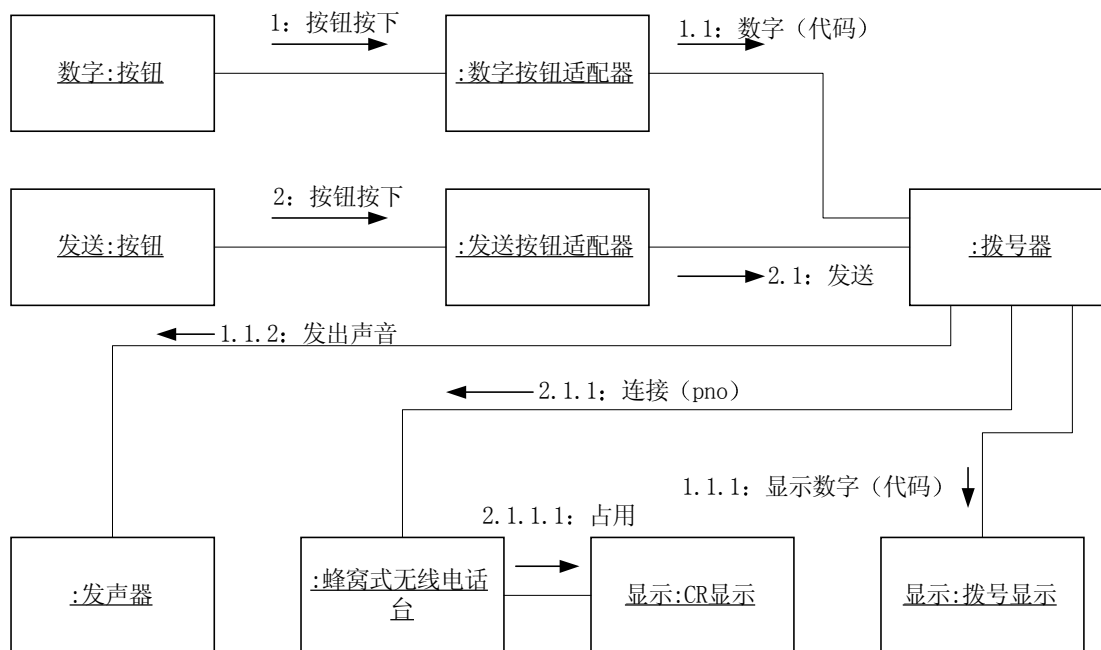


图 1：蜂窝电话的协作图

与这个模型相对应的顺序图显示在图 2 中。图 2 的内容很好理解，尤其在我们将它和图 1 对照时。让我们跟踪这些特性。

首先，现在有两个顺序图。第一个，捕获当数字键被按下时发生的一系列事件。第二个，捕获当用户按发送键呼叫时发生的事件。在每一个图的顶部我们看到表示对象的矩形。就像在协作图中一样，对象名被加下划线以便和它们的类相区分。对象名也被一个冒号将其和类名分开。一些对象，象拨号器，没有特别的对象名，因此冒号作为类名的前缀而没有对象名在它前面。

每个对象向下生出一条虚线，被称为“生命线”。这些线定义了图的时间轴。按常规，时

间延向下方向流逝。¹“生命线”描述了它们所联系的对象将存在多久。在图 2 中我们不能看到“生命线”的整个长度。它们从图的最顶部延伸过图的最低部。这暗示了图中描述的对象存在于图开始之前，也仍存在于图结束之后。

生命线之间的箭头表示正在对象间传递的消息。顺序号，象上一期所显示的，是允许的但并不需要。结束箭头的白色窄条被称为“激活”。它们显示一个响应消息的“方法”的执行期。这些“方法”暗含了“激活”结束后返回到调用者。²

在第一个顺序图中围绕消息组的大矩形定义了一个迭代。这迭代的循环条件被显示在矩形的底部。

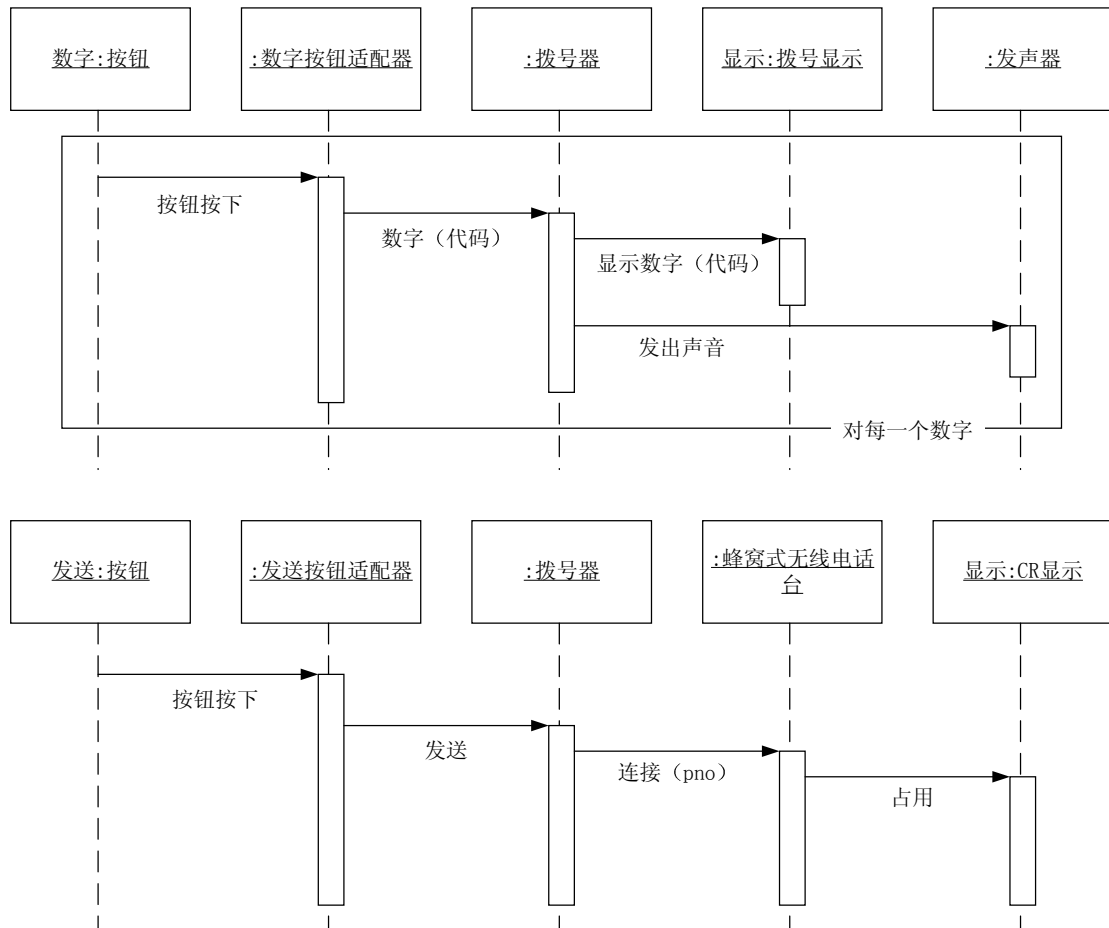


图 2 蜂窝电话的顺序图

花些时间观察图 1 和图 2。你将会亲自证明它们表示了相同的信息。然而两者的形式是根本不同的。顺序图占据了更多的空间，但更容易进行规则系统地跟随。协作图显示整个协作在一个密集的图中，但规则系统有一定的模糊。使用哪一种图依赖于你想强调的信息。有时候你想显示一个协作的“黏着性”，又有时候你想显示规则系统的“流动性”。

对象的建立和删除

图 3 显示了在顺序图中我们如何描述建立和删除一个对象。下面我们看一下“蜂窝式无线电话台”对象建立一个“连接”对象在响应一个连接消息时。建立被一个终止于对象框的箭头所标示。删除同样被一个终止与对象“生命线”的末端 X 处的箭头所标示。

¹ UML 也允许在图的边上提示，让时间向右延续。然而这是很少有的情况。

² 这样的返回能被一个无标注的箭头所显示，从“激活”的底部返回调用者的“生命线”。在异步消息的情况下，一个“激活”的结束不隐含一个返回。

这记号是非常直观的。连接对象的生命期是显而易见的。象这样一些事物是很难在协作图中表示清楚的。

异步消息和并行

注意一下图 3 中一些箭头的头部是不完整的。这“半箭头”标注了异步消息。异步消息是一种在接收对象中产生一条新执行线后就立即返回的消息。³连接消息，例如，立即返回到“蜂窝式无线电话台”对象。然而，你能看到在“连接”对象的“生命线”上的“激活”框，那么连接方法仍在继续执行。这连接方法正在一个分离的线程中执行。这演示了顺序图具有显示并发、多线程交互的能力。在协作图中描述这种信息就要笨拙的多。

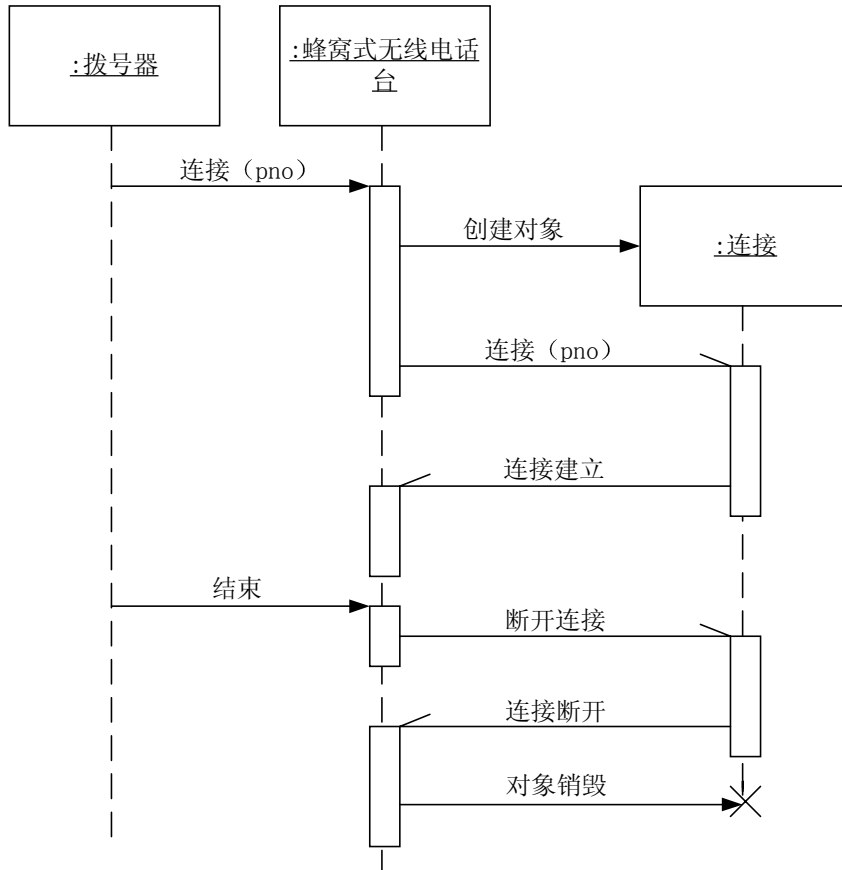


图 3 连接和断开连接

竞争条件

有并行的地方就可能存在竞争条件。竞争条件出现在一个单线程或对象收到来自两个竞争源的消息。如果处理不好，参与对象可能被弄糊涂。考虑图 4。

我们又看到两个顺序图，在这两个图中为了清晰，已将“激活”框忽略。⁴第一个图显示了正常的事件过程，当一个蜂窝电话收到一个呼叫。蜂窝式无线电话台对象检测收到的呼叫并响铃。同时也告诉拨号器对象有一个呼叫正过来。这使得拨号器对象处在一个特殊的状态。当拨号器对象处在这特殊状态时按下发送按钮，那么拨号器对象向蜂窝式无线电话台对象发应答消息，因此呼叫被连接。

³ 事实上并不需要产生新的线程，所需要的是接收对象在一个分离的线程中执行方法。这线程可能存在于异步消息到来之前，而仅仅是等待着做一些事。

⁴ UML 允许设计者根据判断进行这样的忽略。

为什么需要特殊状态？回头看一下图 1。你能看到拨号器对象能接收发送消息，在用户输入电话号码后。因此有两种情况，哪一种情况下拨号器接收发送消息。一是当呼叫时，还有当应答时。当呼叫时，拨号器送连接消息到蜂窝式无线电话台。当应答时，拨号器送应答消息到蜂窝式无线电话台。

这第二个顺序图显示了竞争条件。在这种情况下用户正在拨号。一个呼叫进入蜂窝式无线电话台对象，正在这时用户按下了发送按钮发出一个呼叫。这进入的呼叫消息和连接消息交叉而过。消息的交叉显示了竞争。

当一个消息向下倾斜斜这两个一样，表示在消息的发送和消息的接收之间有一段时间差。在系统中，消息通过网络，很显然消息需要时间来传递，或在队列中等待被接收。在多线程应用中，对象间的消息经常被转换成消息对象等待在队列中。因此，大量的时间可能化在消息的发送（队列）和它们的最终接收和执行中。

明显的，如果蜂窝式无线电话台对象不曾被写下期望事件的返回，它将被搞糊涂。有可能，设计蜂窝式无线电话台对象的状态机的工程师不曾期望到一个连接消息将被接收在它刚刚送出“收到呼叫”消息后。

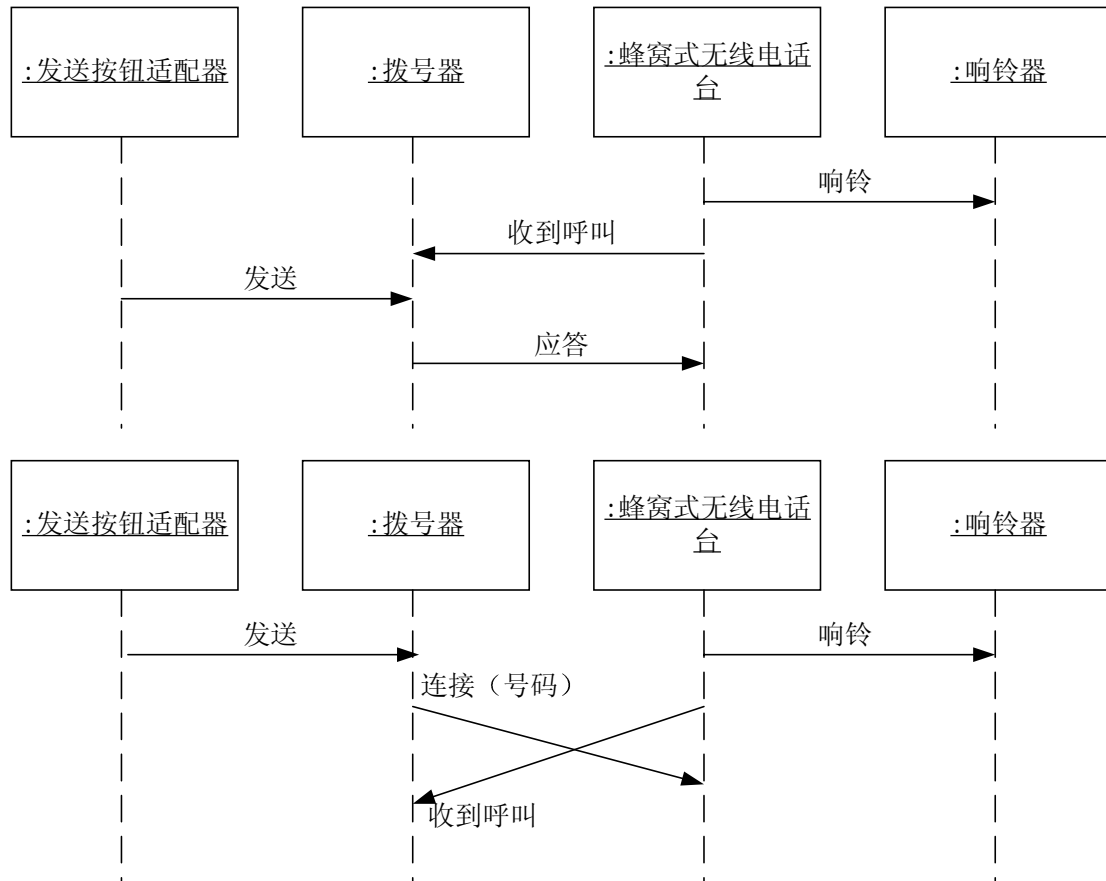


图 4 竞争条件在拨号和应答之间

象这样的竞争条件是一种重要的错误来源在并发系统中。调试它们可能是非常困难，因为它们依赖几个事件的精确定时。这些问题导致“间歇的不可解释和不可重复的崩溃”，这是所有时实软件工程师的祸端。带倾斜箭头的顺序图⁵是一种很好的工具，用来发现哪里会出现竞争条件。

结论

⁵ 这样的图也常被称为“消息顺序图”

很显然，顺序图有诸多优点。它们清楚地描述了事件的顺序，显示了什么时候对象被建立和销毁，在表示并发操作上也很优秀，而更重要的是捕获竞争条件。然而，虽然有很多优点，它们也不是完美的工具。它们太占空间，并且也不能很好地表示协作对象间的相互关系。尽管顺序图是强有力的，我仍然认为密集而简洁的协作图更令人愉快。因此，作为一个个人品味的问题，我仅在需要顺序图的地方使用顺序图。