

34 我们一起学活动图

更新时间：2020-06-03 16:08:23



“读一本好书，就是和许多高尚的人谈话。——歌德”

1. 前言

前一节我们学到了描述对象状态随着事件变化的状态机图，那么有没有一种图描述对象协作关系的图形呢？

答案是，有，那就是活动图。

【强制】如果系统中超过 2 个对象之间存在协作关系，并且需要表示复杂的处理流程，使用活动图来表示。

说明：活动图是流程图的扩展，增加了体现协作关系的对象泳道，支持表示并发等。

那么我们思考几个问题：

- 什么是活动图？
- 为什么要使用活动图？
- 如何画活动图呢？

本小节将为你分析这些问题。

2. 活动图是什么？它的使用场景有哪些？

2.1 活动图的概念

活动图是工作流的图形表示，活动图主要由活动和动作构成，可以支持分支选择、迭代、并行。在 UML 中，活动图主要用于为计算性和组织性过程（即工作流）建模，相关活动之间的数据流也在其覆盖范围。

活动图显示连接动作和其他节点（如决策、分支、连接、合并和对象节点）的流。一般情况下活动和活动图之间是一一对应关系。

除非一个活动图表示一个连续的循环，否则，活动图应该有一个使活动开始的初始动作，还应该有一个或者多个终止动作。实心圆表示活动开始，牛眼符号表示活动结束。

流可以分支和合并。在活动图中用钻石表示分支条件，分支条件的出口由事件（如 **Yes, No**）或守护条件（如下图中的 **[order accepted]**, **[order rejects]**）来控制。

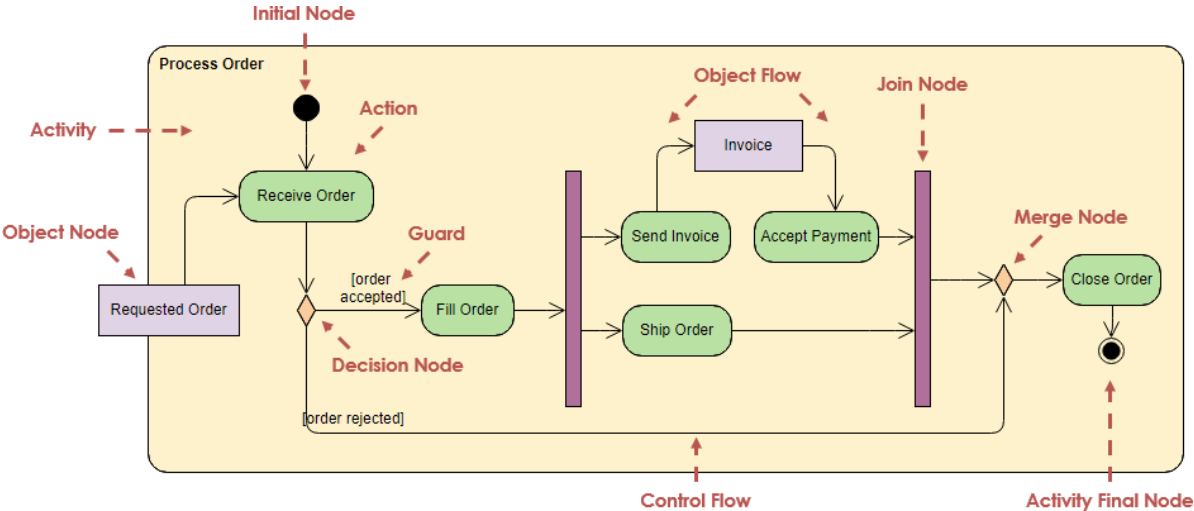


图 1：活动图描述（图片来自 visual paradigm）

流还可以分叉和再连接。这就产生了并发（并行）的计算线程。流的分叉和连接用短线表示。没有并发过程的流程图和传统的流程图非常相似。

2.2 活动图和状态图关系

活动图简化了流程图并添加了一些新的符号。

状态图着重描述从一个状态到另一个状态的流程，主要有外部事件的参与。

活动图将流程分为一个一个活动，通过活动的先后顺序来展示流程；而状态机图从某个事物的状态是如何变化的角度来描述流程。

2.3 使用场景

正如《手册》所说，当系统中超过 2 个对象之间存在协作关系，并且需要保湿复杂的处理流程时，需要使用活动图表示。

3. 绘图

3.1 核心组件

3.1.1 初始节点（Initial Node）

活动发生前的状态称之为开始状态。除非有嵌套活动，否则一个流程只能有一个开始状态。

3.1.2 最终节点（Final Node）

最终节点采用大圆套小圆表示，其中内部的小圆为实心。活动图只能有一个初始状态，但是可以有 0 个或多个最终状态。

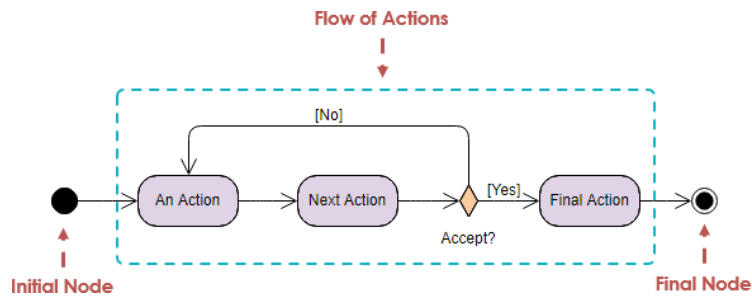


图 2：活动图（图片来自 visual paradigm）

3.1.3 流的最终节点

UML 2.0 新增了一个节点类型，称之为流的最终节点，用来代替活动的最终节点来表示流的终结。

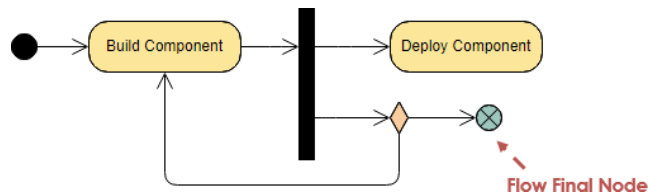


图 3：活动图最终节点（图片来自 visual paradigm）

3.1.4 流程转换

活动图包含多种活动状态，那么这些状态之间通过什么关联呢？状态流转就应运而生了。

状态流转包括控制流转和对象流转。

控制流转（Control Flow） 或状态流转也叫做路径或边。它用来表示从一个活动状态到另外一个活动状态的转变。我们使用带箭头的实线来表示。

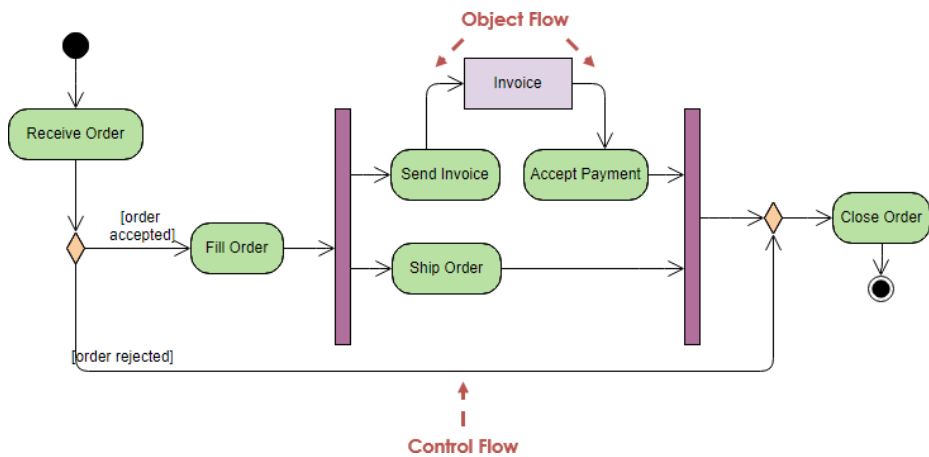


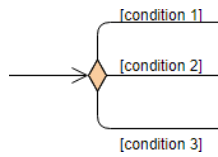
图 4：流程转换（图片来自 visual paradigm）

对象流转（Object Flow） 发生在活动和对象之间。一个活动状态使用对象作为输入，则从对象用箭头指向该活动状态。如果一个活动状态需要更新或者产生一个对象作为输出，那么箭头需要从该活动状态指向对象。

3.1.5 决策节点和分支

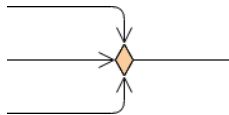
决策节点

决策节点承接一个流程控制来源，将其拆分为多个流程控制出口。



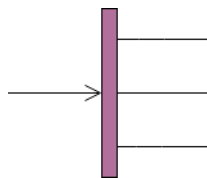
合并节点

合并节点将多个可选分支汇聚到一个节点中。



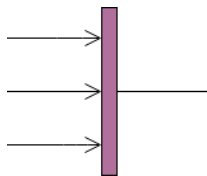
Fork 节点

Fork 节点将一个流程分成多个并发流程。



Join 节点

一个 join 节点是同步多流程的控制节点。它有多条入口边，只有一条出口边。

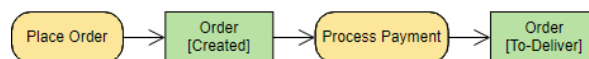


3.1.6 守护

活动图中，守护（Guard）是一种真假条件，决定状态的流转。

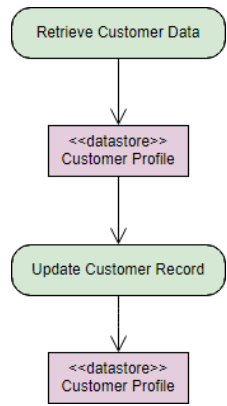
3.1.7 对象节点

UML 2.0 的活动建模支持对象节点。



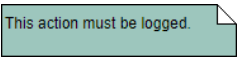
3.1.8 数据存储

数据存储表示对象的持久化。



3.1.9 备注

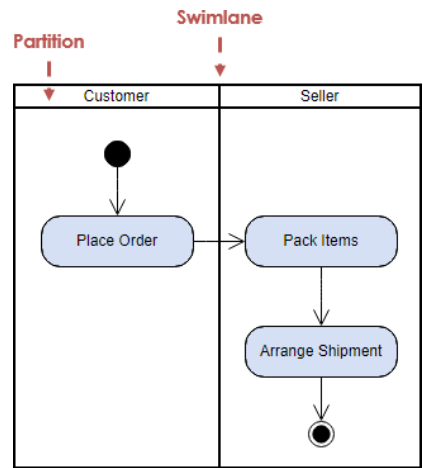
备注支持为活动图中的元素进行注释，可以承载对建模有用的信息。



3.1.0 泳道

泳道表示不同的信息种类，将整个流程图结合到不同得到参与者视角中。

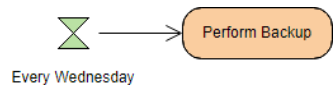
下图表示顾客在线采购超时商品并下单，销售商实地挑选商品并安排运输的泳道图。



3.1.11 时间事件和事件信号

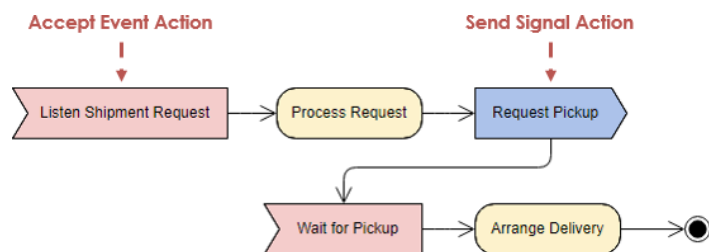
时间事件表示活动的时间描述，采用水漏的图形表示。

下图表示每周三执行备份活动。



接收事件活动（Accept Event Action）在活动图的业务建模中非常重要。它表示接收活动等待事件的发生。事件接收后，活动将被执行。

发送信号活动（Send Signal Action）表示接收活动作出反应的信号。



3.2 绘图步骤

构造结构图通常先为用例添加开始和结束点，为用例的主要步骤添加一个活动，从每个活动到其他活动、决策点和终点添加转换，在并行活动的地方添加同步条。

绘制活动图的主要步骤如下：

1. 首先，决定是否采用泳道，主要根据活动图中是否要体现出活动图的不同实施者；
2. 然后，尽量使用分支、分叉和汇合等基本的构建元素来描述活动控制流程；
3. 如果需要，加入对象流以及对象的状态变化；
4. 如果需要，使用高级建模元素（如辅助活动图、汇合描述、发送信号和接收信号和备注等）来表示更详细的信息。

回顾之前的几种图形的绘制过程，我们可以发现，绘图的步骤基本都是先绘制主要信息再进行丰富，符合先整体后局部，先易后难的方式。因此我们在绘图过程中，不需要背诵具体的绘图步骤，记住这个绘图原则即可。

3.3 参考范例

上一节我们使用状态图绘制了某博客系统发表评论的步骤，本节将使用 [PlantUML](#) 绘制对应的活动图。

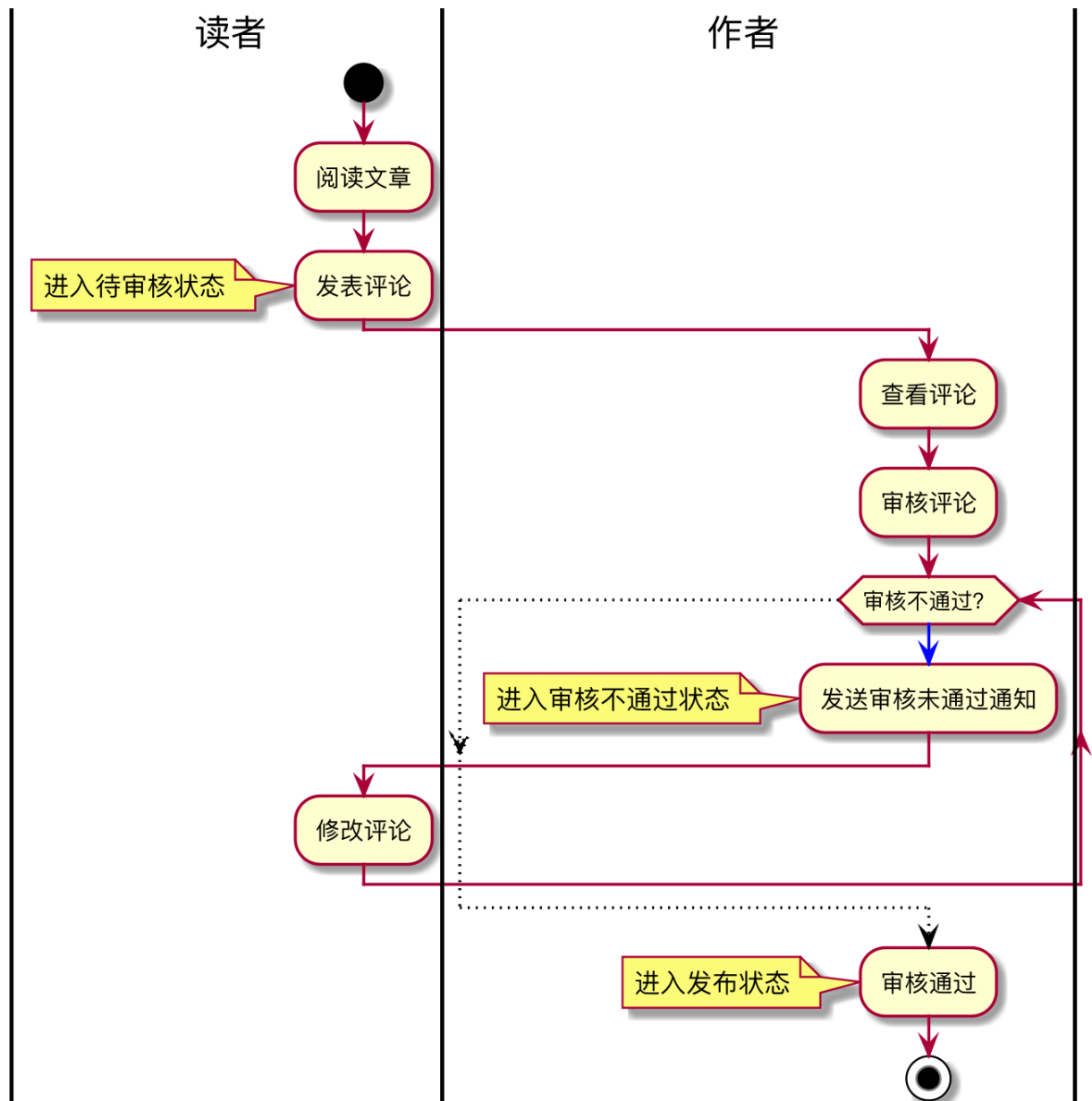
场景描述：

读者在某博客系统阅读文章后可以发表评论，但是评论需要作者审核，审核通过后对其他读者才可见，审核不通过则需要重新修改。

由于这里有两个角色，读者和作者，因此我们采用泳道的方式绘图。

评论审核是否通过需要走不通的流程，因此需要使用分支进行活动的流程控制。

根据场景描述以及上述分析，我们绘图如下：



4. 总结

本节主要介绍了活动图的概念、活动图的使用场景、活动图的核心组件，并给出了活动图的使用范例。希望大家可以结合 PlantUML 中活动图的相关语法示例，结合 visual paradigm 的活动图范例，结合本节给出的绘图步骤，进行模仿绘图。

下一章为避坑篇，讲述开发中常见的坑以及如何才能更好地避坑。

参考资料

阿里巴巴与 Java 社区开发者.《Java 开发手册 1.5.0》华山版. 2019

活动图 - 维基百科

[活动图的语法和功能](#)

[澳] Leszek A. Maciaszek. 《需求分析与系统设计》 第三版. [译] 马素霞。王素琴。谢萍 等。机械工业出版社. 2019

谭云杰. 《大象：Thinking in UML》. 中国水利水电出版社. 2012

张传波. 《火球：UML 大战需求分析》. 中国水利水电出版社. 2012

}