uml活动图的概念与作用

一．UML概述

以下内容对uml进行简单介绍，读者有兴趣可以阅读，建议读者可以阅读其他uml等文章，最后有时间在了解这个部分内容。这样可以先去实践一些uml，然后在回到这里的一些简单的理论介绍，收获会大些。

UML 全称Unified Modeling Language 又称统一建模语言或标准建模语言,是始于1997年一个OMG标准，它是一个支持模型化和软件系统开发的图形化语言，为软件开发的所有阶段提供模型化和可视化支持。

UML是一种定义良好、易于表达、功能强大且普遍适用的建模语言。它融入了软件工程领域的新思想、新方法和新技术。它的作用域不限于支持面向对象的分析与设计，还支持从需求分析开始的软件开发的全过程。

作为一种建模语言，UML的定义包括UML语义和UML表示法两个部分。

(1) UML语义 描述基于UML的精确元模型定义。元模型为UML的所有元素在语法和语义上提供了简单、一致、通用的定义性说明，使开发者能在语义上取得一致，消除了因人而异的最佳表达方法所造成的影响。此外UML还支持对元模型的扩展定义。

(2) UML表示法 定义UML符号的表示法，为开发者或开发工具使用这些图形符号和文本语法为系统建模提供了标准。这些图形符号和文字所表达的是应用级的模型，在语义上它是UML元模型的实例。

标准建模语言UML的重要内容可以由下列五类图（共9种图形）来定义：

第一类是用例图，

从用户角度描述系统功能，并指出各功能的操作者。

第二类是静态图 (Static diagram)，

包括类图、对象图和包图。其中类图描述系统中类的静态结构。不仅定义系统中的类，表示类之间的联系如关联、依赖、聚合等，也包括类的内部结构（类的属性和操作）。类图描述的是一种静态关系，在系统的整个生命周期都是有效的。

对象图是类图的实例，几乎使用与类图完全相同的标识。他们的不同点在于对象图显示类的多个对象实例，而不是实际的类。一个对象图是类图的一个实例。由于对象存在生命周期，因此对象图只能在系统某一时间段存在。

包由包或类组成，表示包与包之间的关系。包图用于描述系统的分层结构。

第三类是行为图（Behavior diagram），

描述系统的动态模型和组成对象间的交互关系。行为图包括：状态图、活动图、顺序图和协作图。其中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。通常，状态图是对类图的补充。在实用上并不需要为所有的类画状态图，仅为那些有多个状态其行为受外界环境的影响并且发生改变的类画状态图。 而活动图描述满足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。活动图是一种特殊的状态图，它对于系统的功能建模特别重要，强调对象间的控制流程。 顺序图展现了一组对象和由这组对象收发的消息，用于按时间顺序对控制流建模。用顺序图说明系统的动态视图。 协作图展现了一组对象，这组对象间的连接以及这组对象收发的消息。它强调收发消息的对象的结构组织，按组织结构对控制流建模。 顺序图和协作图都是交互图，顺序图和协作图可以相互转换。

第四类是交互图（Interactive diagram），

描述对象间的交互关系。其中顺序图显示对象之间的动态合作关系，它强调对象之间消息发送的顺序，同时显示对象之间的交互；合作图描述对象间的协作关系，合作图跟顺序图相似，显示对象间的动态合作关系。除显示信息交换外，合作图还显示对象以及它们之间的关系。如果强调时间和顺序，则使用顺序图；如果强调上下级关系，则选择合作图。这两种图合称为交互图。

第五类是实现图 ( Implementation diagram )。

其中构件图描述代码部件的物理结构及各部件之间的依赖关系。一个部件可能是一个资源代码部件、一个二进制部件或一个可执行部件。它包含逻辑类或实现类的有关信息。部件图有助于分析和理解部件之间的相互影响程度。

配置图定义系统中软硬件的物理体系结构。它可以显示实际的计算机和设备（用节点表示）以及它们之间的连接关系，也可显示连接的类型及部件之间的依赖性。在节点内部，放置可执行部件和对象以显示节点跟可执行软件单元的对应关系。

从应用的角度看，当采用面向对象技术设计系统时，

首先是描述需求；

其次根据需求建立系统的静态模型，以构造系统的结构；

第三步是描述系统的行为。

其中在第一步与第二步中所建立的模型都是静态的，包括用例图、类图（包含包）、对象图、组件图和配置图等五个图形，是标准建模语言UML的静态建模机制。

其中第三步中所建立的模型或者可以执行，或者表示执行时的时序状态或交互关系。它包括状态图、活动图、顺序图和合作图等四个图形，是标准建模语言UML的动态建模机制。因此，标准建模语言UML的主要内容也可以归纳为静态建模机制和动态建模机制两大类。

二．活动图介绍

1. 简介

活动图是uml的动态模型的一种图形，一般用来描述相关用例图。准确的活动图定义：活动图描述满足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。活动图是一种特殊的状态图，它对于系统的功能建模特别重要，强调对象间的控制流程。

交互图强调的是对象到对象的控制流，而活动图则强调的是从活动到活动的控制流

活动图是一种表述过程基理、业务过程以及工作流的技术。

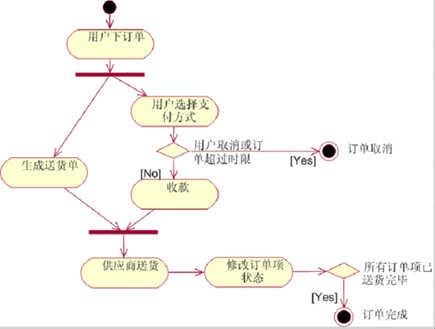
它可以用来对业务过程、工作流建模，也可以对用例实现甚至是程序实现来建模

2. 活动图示例

下图是个简单的活动图例子，里面包括了大部分活动图的相关元素，大家应该都能看的差不多吧，有不明白的可以继续看，下面有针对各个元素有介绍啦，相信大家看完后面的，看这个图就不会有问题的。

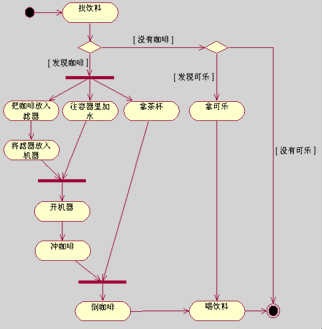
另外，若想学会画活动图，必须先看大量的图，看明白别人的图，在慢慢画，慢慢一个图形就弄明白了。

其实uml包括了各种各样的图形，把每种图形都会画啦，基本uml也就会啦。



3. 活动图与流程图的区别

活动图描述系统使用的活动、判定点和分支，看起来和流程图没什么两样，并且传统的流程图所能表示的内容，大多数情况下也可以使用活动图表示，但是两者是有区别的，不能将两个概念混淆。



活动图与流程图的区别

⑴ 流程图着重描述处理过程，它的主要控制结构是顺序、分支和循环，各个处理过程之间有严格的顺序和时间关系

活动图描述的是对象活动的顺序关系所遵循的规则，它着重表现的是系统的行为，而非系统的处理过程。

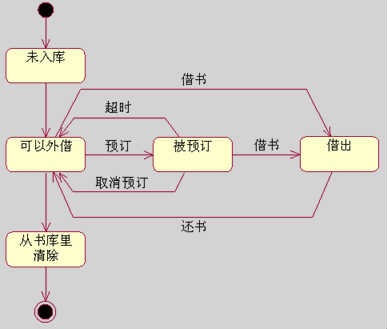
⑵ 活动图能够表示并发活动的情形，而流程图不能。

⑶ 活动图是面向对象的，而流程图是面向过程的。

4. 活动图与状态图区别

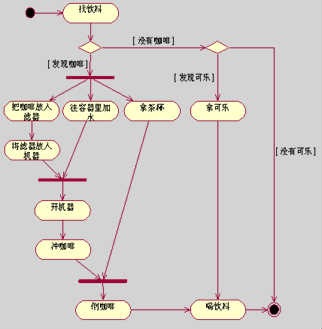
活动图与状态图都是状态机的表现形式，但是两者还是有本质区别：

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。



上图是一个典型的状态图

活动图着重表现从一个活动到另一个活动的控制流，是内部处理驱动的流程。



5. 活动图基本元素

1) 初始节点和活动终点：

实心圆表示初始节点（只有一个），圆圈内加一个实心圆来表示活动终点（可有多个）。wps_clip_image-11674[3][1]

2) 活动节点：

用来表示一个活动

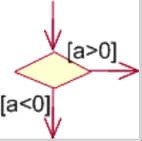


3) 转换：

一条带箭头的直线来表示。 一旦前一个活动结束马上转到下一个活动（无触发转换）。

4) 分支与监护条件：

分支是用菱形表示的，它有一个进入转换（箭头从外指向分支符号），一个或多个离开转换（箭头从分支符号指向外）。而每个离开转换上都会有一个监护条件，用来表示满足什么条件的时候执行该转换。

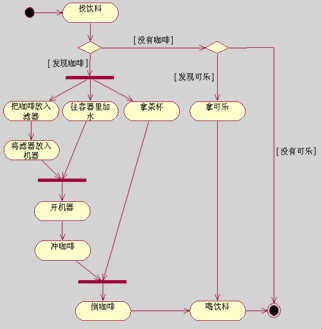


5) 分叉与汇合：

分叉用于将动作流分为两个或者多个并发运行的分支，而汇合则用于同步这些并发分支，以达到共同完成一项事务的目的。

分叉可以用来描述并发线程，每个分叉可以有一个输入转换和两个或多个输出转换，每个转换都可以是独立的控制流。

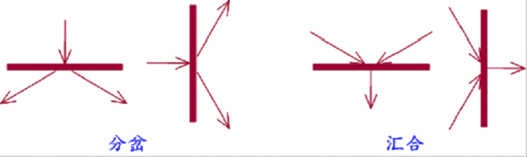
汇合代表两个或多个并发控制流同步发生，当所有的控制流都达到汇合点后，控制才能继续往下进行。



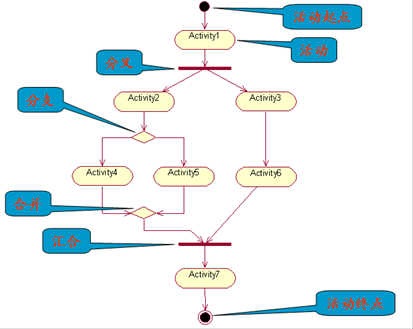
每个汇合可以有两个或多个输入转换和一个输出转换。

汇合将两条路径连接到一起，合并成一条路径。汇合指的是两个或者多个控制路径在此汇合的情况。汇合是一种便利的表示法，省略它不会丢失信息。汇合和分支常常成对的使用，合并表示从对应分支开始的条件行为的结束。

分叉和汇合都使用加粗的水平线段表示。



6. 抽象的活动图示例



UML的活动图中包含的图形元素有动作状态、活动状态、动作流、分支与合并、分叉与汇合、泳道和对象流等。

7. 带泳道的活动图

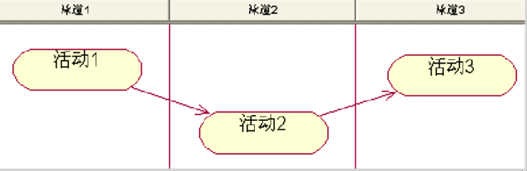
泳道表明每个活动是由哪些人或哪些部门负责完成。



每个泳道代表特定含义的状态职责的部分。在活动图中，每个活动只能明确的属于一个泳道，泳道明确的表示了哪些活动是由哪些对象进行的。

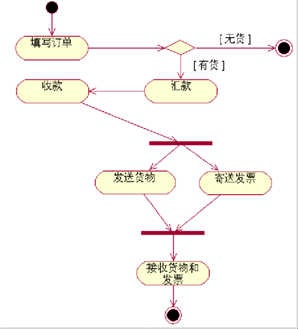
每个泳道都有一个与其他泳道不同的名称。

每个泳道可能由一个或者多个类实施，类所执行的动作或拥有的状态按照发生的事件顺序自上而下的排列在泳道内。

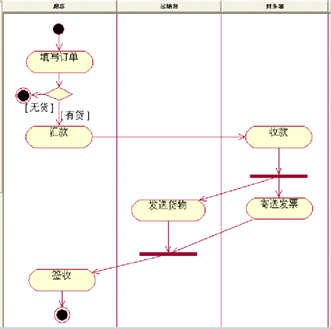


在活动图中泳道区分了负责活动的对象，它明确地表示了哪些活动是由哪些对象进行的。

在包含泳道的活动图中每个活动只能明确地属于一个泳道



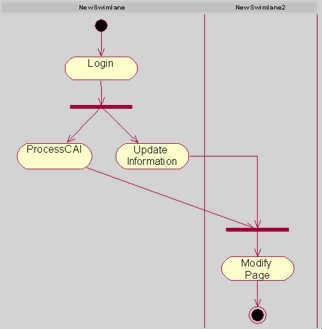
上图是一个未采用泳道的活动图



上图是采用泳道技术后的活动图

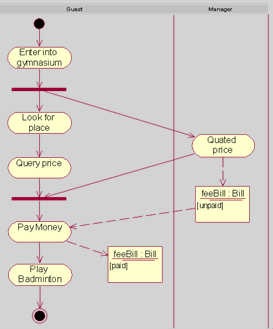
从两幅图的对比中，我们可以了解泳道技术是非常重要的，可以更明确表达出活动图的意图。

泳道将活动图中的活动化分为若干组，并把每一组指定给负责这组活动的业务组织，即对象。



泳道区分了负责活动的对象，它明确地表示了哪些活动是由哪些对象进行的。

在包含泳道的活动图中，每个活动只能明确地属于一个泳道。



在活动图中，泳道用垂直实线绘出，垂直线分隔的区域就是泳道。

在泳道上方可以给出泳道的名字或对象(对象类)的名字，该对象(对象类)负责泳道内的全部活动。

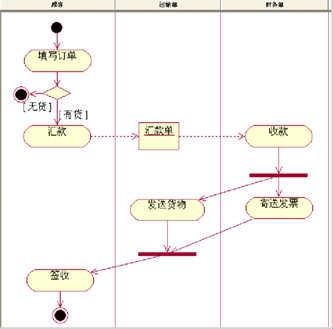
泳道没有顺序，不同泳道中的活动既可以顺序进行也可以并发进行。

动作流和对象流允许穿越分隔线。

8. 带对象流的活动图

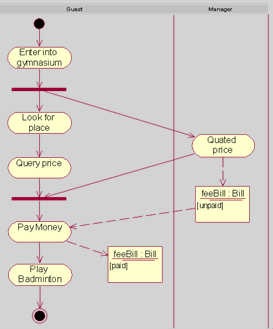
用活动图描述某个对象时，可以把涉及到的对象放置在活动图中，并用一个依赖将其连接到进行创建、修改和撤销的动作状态或者活动状态上，对象的这种使用方法就构成了对象流。

wps_clip_image-30329[3][1]



对象流是动作状态或者活动状态与对象之间的依赖关系

对象流表示动作使用对象或者动作对对象的影响。



对象流中对象的特点：

⑴ 一个对象可以由多个动作操纵；

⑵ 一个动作输出的对象可以作为另一个动作输入的对象；

对象流中对象的特点：

⑶ 在活动图中，同一个对象可以多次出现，它的每一次出现表明该对象正处于对象生存期的不同时间点。

在活动图中，对象流用带有箭头的虚线表示。

如果箭头从动作状态出发指向对象，则表示动作对对象施加了一定的影响。

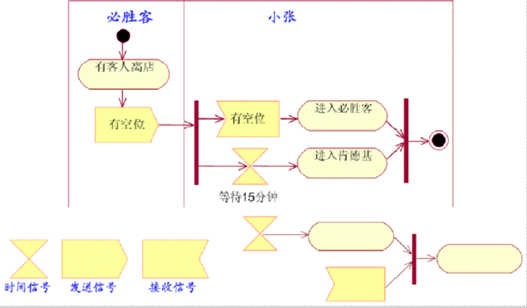
施加的影响包括创建、修改和撤销等。如果箭头从对象指向动作状态，则表示该动作使用对象流所指向的对象。

状态图中的对象用矩形表示，矩形内是该对象的名称，名称下的方括号表明对象此时的状态。

还可以在对象名称的下面加一个分隔栏表示对象的属性值。

9. 信号发送和接收

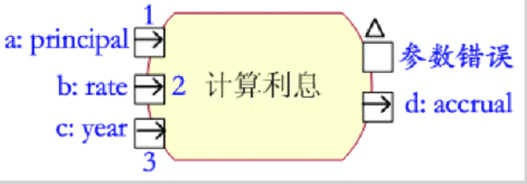
发送信号与接收信号



10. 引脚

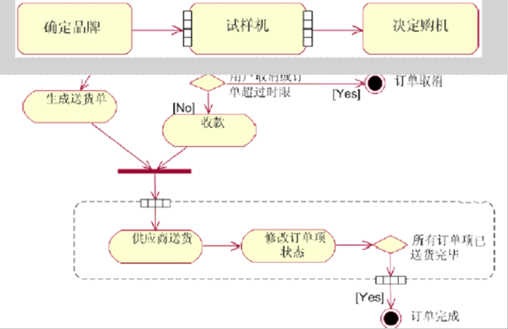
是一个对象节点，代表活动连接输入、输出值的连接点

用来标明每个活动节点所需输入的数据或者所产生的数据（建模业务流时则可表示产生或者消耗的资源）



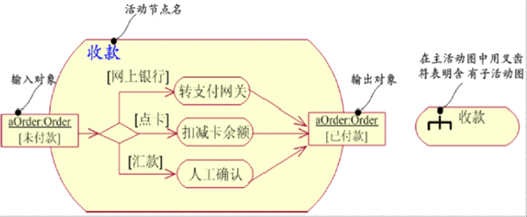
11. 扩展区

表示重复或循环



12. 辅助活动图

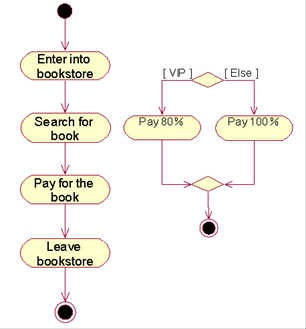
当活动图过于复杂时可以用活动的分解来处理

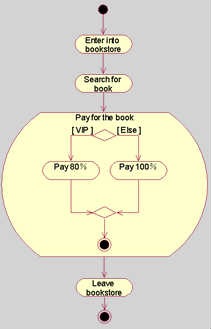


一个活动可以分为若干个动作或子活动，这些动作和子活动本身又可以组成一个活动图。

不含内嵌活动或动作的活动称之为简单活动；

嵌套了若干活动或动作的活动称之为组合活动，组合活动有自己的名字和相应的子活动图





三．活动图绘制要点

⑴ 识别要对工作流描述的类或对象。找出负责工作流实现的业务对象，这些对象可以是显示业务领域的实体，也可以是一种抽象的概念和事物。找出业务对象的目的是为每一个重要的业务对象建立泳道。

⑵ 确定工作流的初始状态和终止状态，明确工作流的边界。

⑶ 对动作状态或活动状态建模。找出随时间发生的动作和活动，将它们表示为动作状态或活动状态。

⑷ 对动作流建模。对动作流建模时可以首先处理顺序动作，接着处理分支与合并等条件行为，然后处理分叉与汇合等并发行为。

⑸ 对对象流建模。找出与工作流相关的重要对象，并将其连接到相应的动作状态和活动状态。

⑹ 对建立的模型进行精化和细化。