# 9.1 包图[2]

包图描绘模型元素在包内的组织和依赖关系，包括包的导入和包扩展。它们还提供相应命名空间的可视化。

## 9.1.1概述

包是一种把元素组织到一起的通用机制，包可以嵌套于其他包中。

包图用于描述包与包之间的关系，包的图标是一个带标签的文件夹。



## 9.1.2包之间的关系

**引入关系**：一个包中的类可以被另一个指定包（以及嵌套于其中的那些包）中的类引用。

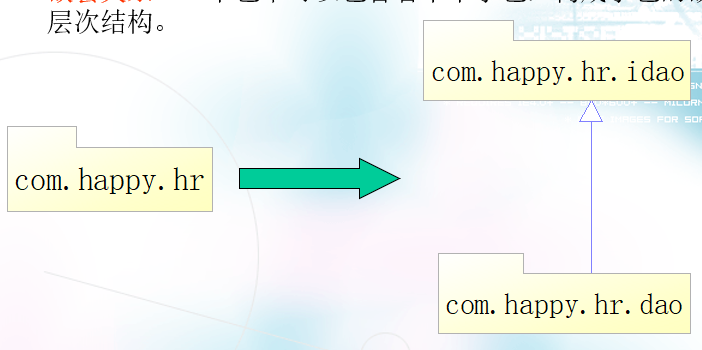
引入关系是依赖关系的一种，需要在依赖线上增加一个<<import>>衍型，包之间一般依赖关系都属于引入关系。



**泛化关系**：表示一个包继承了另一个包的全部内容，同时又补充自己增加的内容。



**嵌套关系**：一个包中可以包含若干个子包，构成了包的嵌套层次结构。



**包图实例（基于C/S的人力资源管理系统）：**

****

# 9.2 结构组合图[1][2]

组合结构图反映类、 接口或组件 （和它们的属性） 来描述功能内部的合作。组合结构图和类图类似，只是它们的模型结构的特定使用。类图建模类的静态结构，包括它们的属性和行为。

## 9.2.1结构组合图概述

组合结构图将每一个类放在一个整体中，从类的内部结构来审视一个类。

组合结构图可用于表示一个类的内部结构。



## 9.2.2结构组合图主要元素

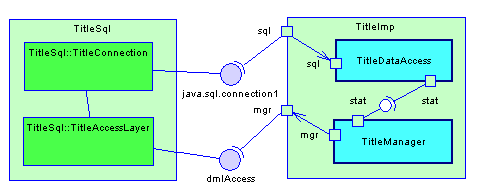
* 部件(Part)：表示被描述事物所拥有的内部成分。
* 连接件(Connector)：表示部件之间的关系。
* 端口(Port)：表示部件和外部环境的交互点。

## 9.2.3组合结构图的作用

* 组合结构图所能够表达的信息，使用组合或者聚合也能够表示，只是一种新的表达形式。
* 组合结构图可以表示一个类的内部成员对象之间的相互关系，是对传统类图的一个补充。
* 组合结构图适用于表示含有内部类的类与外部接口之间的相互关系。

实例：

TitleSql类包含两个内部类，TitleImp类包含两个部件，它们之间通过接口dmlAccess和java.sql.connection1产生连接。



# 9.3 定时图[2]

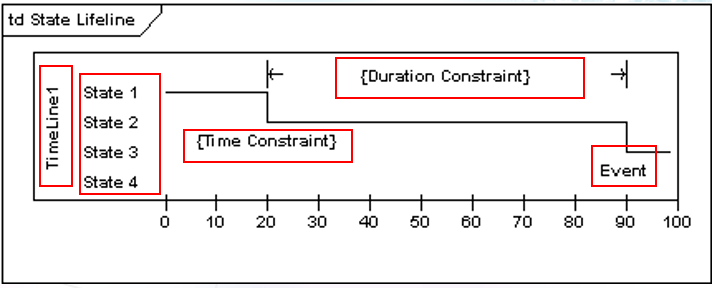
定时图可以把状态发生变化的时刻以及各个状态所持续的时间具体地表示出来。如果把多个对象放在一个定时图中，还可以把它们之间发送和接收消息的时刻表示出来。在这方面，定时图与其他几种交互图相比具有独到的优势。

## 9.3.1定时图定义

1. 定时图采用一种带数字刻度的时间轴来精确地描述消息的顺序，而不是像顺序图那样只是指定消息的相对顺序，而且它还允许可视化地表示每条生命线的状态变化，当需要对实时事件进行定义时，定时图可以很好地满足要求。
2. 定时图的焦点集中于生命线内部以及它们之间沿着时间轴的条件变化。
3. 定时图可以把状态发生变化的时刻以及各个状态所持续的时间具体地表示出来。如果把多个对象放在一个定时图中，还可以把它们之间发送和接收消息的时刻表示出来。在这方面，定时图与其他几种交互图相比具有独到的优势。
4. 定时图来自于电子工程领域，在需要明确定时约束一些事件时可以使用它们。

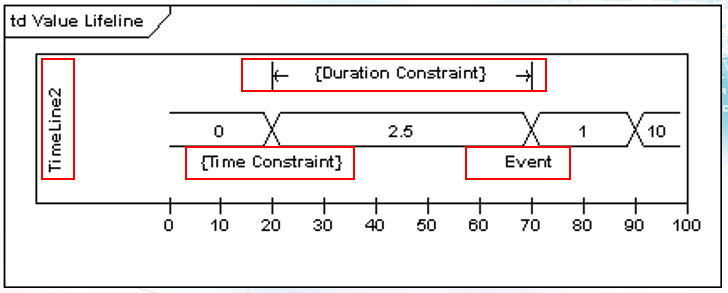
## 9.3.2定时图组成

状态的变化



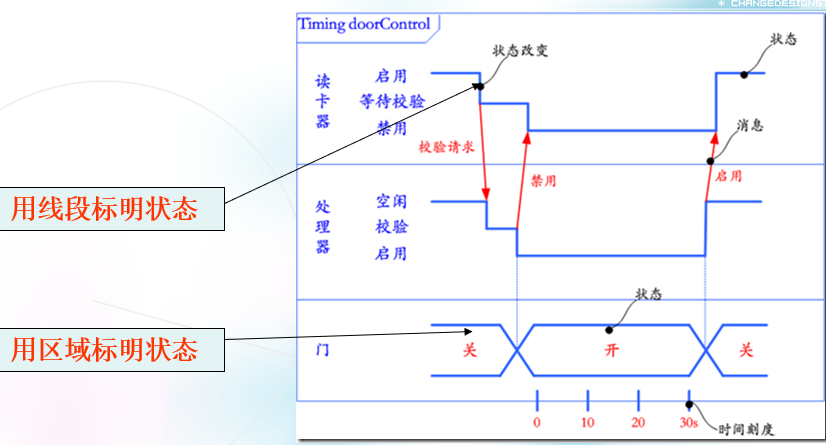
最左边是对象，State表示状态，Time Constraint表示时间单位，Duration Constraint表示持续时间，转折点表示Event事件。

值的变化



跟状态定时图略有相似，可以参考状态定时图理解。

实例：



## 9.3.3定时图的作用

* 定时图用于表示不同对象上状态改变之间的定时约束，如果需要对交互时间进行控制可使用定时图。
* 对于那些时间指标要求很高或者时序关系复杂而又敏感的系统（如实时系统和通信领域的某些系统）而言，定时图是一种有力的描述手段。
* 在大部分应用系统的建模中，一般不需要用定时图来描述对象的行为以及它们之间的交互，但是可能需要用它描述系统中某些局部对象的交互情况。

# 9.4 交互概览图[2]

交互概览图用于将一些零散的顺序图组织在一起，它采用了活动图的构造方式，利用了活动图的各种控制节点，并把活动图的每个活动结点替换为一个交互或者交互使用。

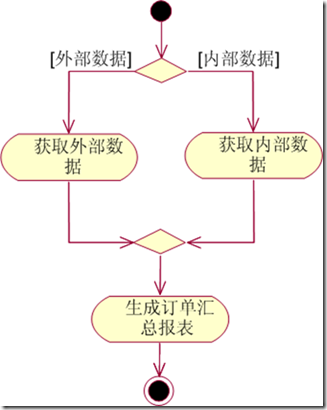
## 9.4.1交互概览图的定义

交互概览图是交互图（包括顺序图，通信图）与活动图的混合物，可以把交互概览图理解为细化的活动图，在其中的活动都通过一些小型的顺序图来表示；也可以将其理解为利用标明控制流的活动图分解过的顺序图。

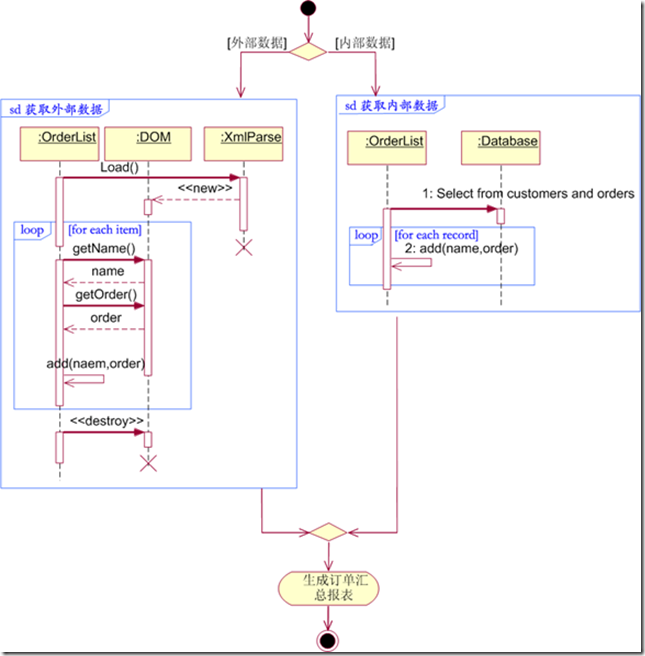
交互概览图用于将一些零散的顺序图组织在一起，它采用了活动图的构造方式，利用了活动图的各种控制节点，并把活动图的每个活动结点替换为一个交互或者交互使用。每个交互或者交互使用都使用一个顺序图表示。

实例：

交互概览图使用活动图表述主线



使用顺序图描述细节



## 9.4.2交互概览图的基本元素

包含了活动图的基本元素：状态，转移，分支，汇合，泳道，对象流

也包含了顺序图的基本元素：角色，对象，生命线，激活期，消息

## 9.4.3交互概览图的作用

* 交互概览图试图将活动图中活动结点之间的控制流机制和顺序图中的生命线间的消息序列混合在一起，很多人认为并没有加入多少新特性。因此，一般情况下很少绘制交互概览图。
* 不要盲目地使用交互概览图，对于规模稍大的场景，它并不是一个很好的选择，它将使模型的可读性大大降低，且不便于文档化。

# UML2.0特性[2][3][4][5]

1. UML 2.0新增加的4种图，主要是作为原有9种的图的扩展内容，实际应用中除了包图，其他三种图的应用较少。但是在一些特殊的环境下有着重要的作用。
2. UML2.0 完全建立在UML1.x基础之上，大多数的UML1.x模型在UML2.0中都可用。但UML2.0在结构建模方面有一系列重大的改进，包括结构类、精确的接口和端口、拓展性、交互片断和操作符以及基于时间建模能力的增强。
3. UML2.0相比较于UML1.x，交互概览图将活动图和顺序图结合在一起，使活动图成为了一系列顺序图的“主框图”。
4. 在UML1.x中，活动图和状态图具有同样的语义。换句话说，它们表达同样的事情，但用了不同的标记。 在UML2.0中， 活动图是基于Petri网络令牌语义， 而不是基于有限状态自动控制，因此更具有表现力。虽然它对计算算法不是很重要，但是对过程建模很有帮助。
5. UML2.0包含了表示结构元素(对象)之间数据流动的流程图(Flow Diagrams)。它与协作图(collaboration diagram)中的信息流动类似，但不完全相同。

发展历史：

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 描述 |
| 1.1 | 1997-11 | OMG采用UML 1.1提案。 |
| 1.3 | 2000-03 | 包含对uml元模型、语义和符号的一些更改，但应被视为对原始提案的一个小升级。 |
| 1.4 | 2001-09 | 主要是“调优”发布，但不完全向上兼容的uml 1.3。将配置文件作为uml扩展包在一起添加。更新特征的可见性。在交互图中插入箭头表示异步调用。模型元素现在可能有多种刻板印象。澄清了合作关系。完善的组件定义和相关概念。工件被添加以表示组件的物理表示。 |
| 1.5 | 2003-03 | 添加操作规范--可执行的动作和过程，包括它们的运行时语义，定义了数据流的概念，以便在动作之间携带数据，等等。 |
| 1.4.2 | 2005-01 | 该版本已被接受为ISO规范（标准）ISO / IEC19501。两年前发布了UML 1.5。 |
| 2.0 | 2005-08 | 新图：对象图、包图、复合结构图、交互概览图、时序图、配置文件图。协作图被重新命名为通信图。  活动图和 序列图 得到了增强。重新设计了活动以使用类似Petri的语义。边缘现在可以包含在分区中。分区可以是分层的和多维的。明确建模的 对象流是新的。  类已经扩展为内部结构和端口（复合结构）。添加了信息流。协作现在是分类器的一种，可以有任何类型的行为描述关联。交互现在包含在分类器中，而不仅仅是协作中。现在可以将用例一般地归为分类器而不仅仅是包。  使用组合片段对并发和分支进行新标记。标记和/或语义被更新用于组件、实现、工件部署。组件不能直接部署到节点。工件应该被部署。已将实现替换为“清单”。工件现在可以显示任何可打包的元素（而不仅仅是组件，如前所述）。现在可以部署到具有内部结构的节点。  添加了新的元类：连接器，协作使用，连接器端， 设备，部署规范， 执行环境，接受事件操作，发送对象操作，结构特征操作，值引脚，活动最终，中央缓冲节点，数据存储，流最终，可中断区域，循环节点，参数， 端口，行为，行为分类器，持续时间，间隔，时间约束，组合片段，创建事件，销毁事件，执行事件，交互片段，交互使用，接收信号事件，发送信号事件，扩展等。  标准UML配置文件中删除了许多陈规定型观念，例如《destroy》，《facade》，《friend》，《profile》，《requirement》，《table》，《thread》。  通过更好地支持可执行模型，改进了结构和行为模型之间的集成。 |
| 2.1 | 2006-04 | 对uml 2.0的小修改-校正和一致性改进。 |
| 2.1.1 | 2007-02 | 对UML 2.1进行了小修订 |
| 2.1.2 | 2007-11 | 对UML 2.11进行了小修改 |
| 2.2 | 2009-02 | 解决了许多小的一致性问题，并对uml2.1.2作了补充说明 |
| 2.3 | 2010-05 | 对UML 2.2进行了少量修订，阐明了 关联 和关联类，添加了 最终分类器，更新了组件图，复合结构，操作等。 |
| 2.4.1 | 2011-08 | UML修订版，很少对类、包进行修复和更新-添加了URI包属性；更新了操作；删除了创建事件、执行事件、发送和接收操作事件、发送和接收信号事件、将销毁事件重命名为销毁发生规范；配置文件-chanGED构造型和应用的构造型具有大写首字母-“元类”和构造型应用。 |
| 2.5 | 2015-06 | uml 2.5被称为对uml 2.4.1的“小修改”，而他们花费了大量的努力来简化和重组uml规范文档。uml规范被重新编写，“以便于阅读”。例如，他们试图“尽可能减少转发引用”。  不再有两个单独的基础结构和上层结构文档，UML 2.5规范是单个文档。 规范中不再使用包合并。  消除了四个UML合规级别（L0，L1，L2和L3），因为它们在实践中没有用。UML 2.5工具必须支持完整的UML规范。 信息流， 模型和 模板 不再是辅助UML构造。同时， 用例， 部署和信息流 成为UML 2.5中的“ 补充概念 ”。  UML 2.5添加了许多修复，说明和解释。他们更新了多重性和多重性元素的描述，阐明了聚合和组合的定义，最后修复了汽车工厂的错误“实例化”依赖性示例。 引入了具有插入符号'^'的继承成员的新表示法 。UML 2.5澄清了 功能重新定义 和重载。他们还移动并重新定义了限定符。  泛化集合的缺省值从{不完整,不相交}改为{不完整,重叠}。  对原型、状态机和活动的澄清和修正很少。协议状态机现在使用“协议”而不是{协议}来表示。不再需要用例来表达行为者的某些需要并由行为者提出。 |

**习题：**

1. 组合结构图与类图有何共同点和区别？它有哪些基本的元素？

答：类图主要显示系统中的类，接口以及它们之间的静态关系；而组合结构图则是将类放在整体中，从类的内部来审视类。

组合结构图的基本元素有：部件，连接件，端口

1. 使用定时图的主要目的是什么？

答：

* 对交互时间进行控制
* 在描绘时间指标要求高或者时序关系复杂敏感的系统时比较方便

1. 交互概览图通常是哪两种图的混合图？为什么提出交互概览图？

答：活动图和交互图。为了结合使用活动图和顺序图，用活动图描述流程，顺序图填充细节。

# 参考资料

[1]《UML用户指南》 人民邮电出版社 Grady Booch, James Rumbaugh, Ivar Jacobson著 邵维忠 麻志毅 马浩海 刘辉 译 2013年1月第1版

[2]《UML2基础、建模与设计教程》 清华大学出版社 杨弘平等 2015年10月第1版

[3] UML2.0与1.x的异同

<https://blog.csdn.net/antony0203/article/details/1966685>

[4] UML简介 <http://www.uml.org.cn/modeler/20190611.asp>

[5] UML 2.5历程 <https://www.cnblogs.com/dragonflyyi/p/3580411.html>