



状态图 State Model

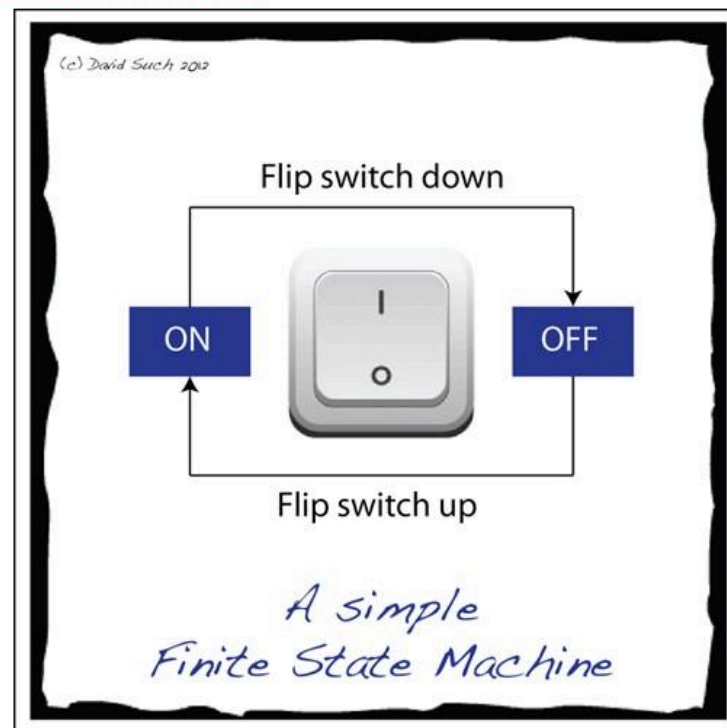
Xu Sihan (徐思涵)
College of Computer Science
Nankai University

*Slides adapted from materials by Prof. Qiang Liu (Tsing Hua University) and
Ivan Marsic (Rugers University)*



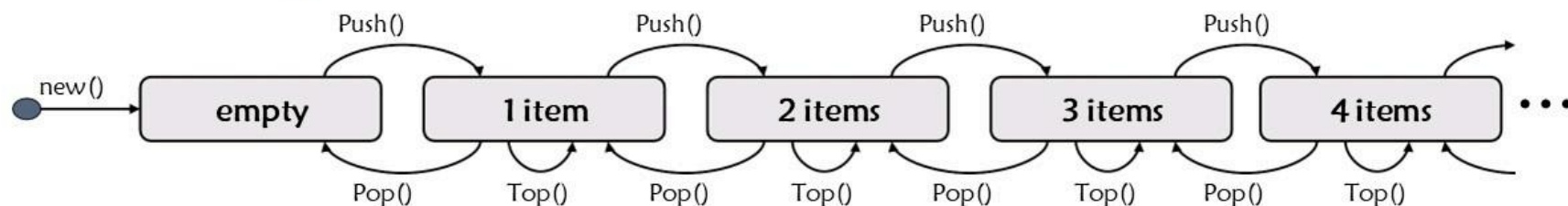
状态建模

- 什么是状态
 - 一个对象的状态空间
 - 具体状态与抽象 状态
- 有限状态机的主要元素
 - 状态和转移
 - 事件和行为
- 模块化的状态机模型：状态图
 - 组合状态和子状态
 - 绘制状态图的方法



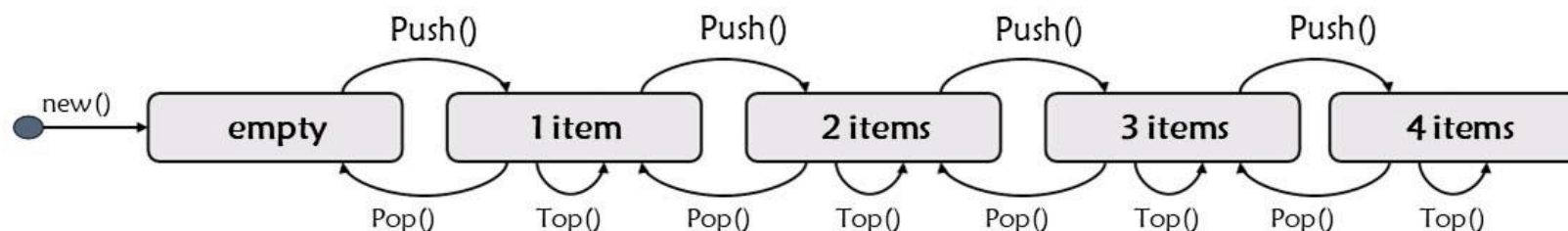
对象及其状态

- 所有的对象都有“状态”
 - 对象存在或者不存在
 - 对象不存在也是一种状态
 - 如果对象存在，则具有相应表示其属性的值
 - 每一种状态表示一种可能的状态赋值
- 例如：栈



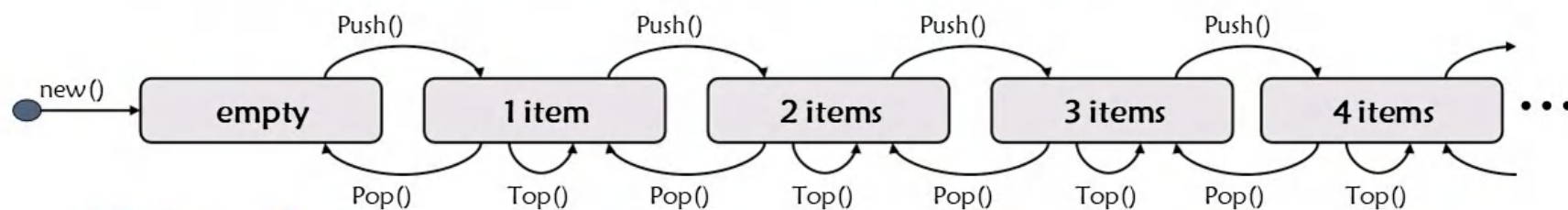
有限状态机

- 有限数量的状态 (所有的属性取值为有限的范围)
 - 例如, 一个最大容量为4的栈
- 模型可以表示动作序列 (状态变化)
 - 例如. `new();Push();Push();Top();Pop();Push()...`
 - 例如 `new();Push();Pop();Push();Pop()...`



状态空间

- 对于大部分对象而言，状态空间是非常庞大的
 - 状态空间大小是对象每个属性取值空间的乘积加1
 - 例如. 具有5个布尔值属性的对象有 2^5+1 个状态
 - 例如. 具有5个整数值属性的对象有 $(\text{maxint})^5+1$ 个状态
 - 例如. 具有5个实数值属性的对象具有?? 个状态
 - 如果忽略计算机表示的局限性，状态空间是无限的

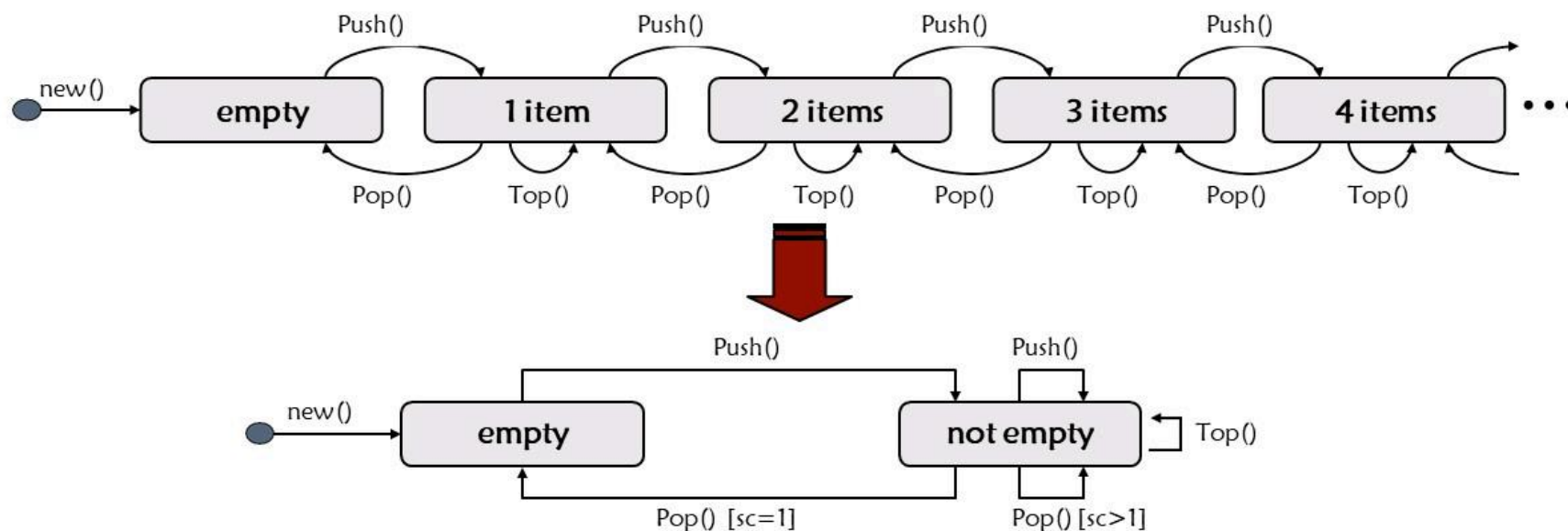


状态的抽象表示

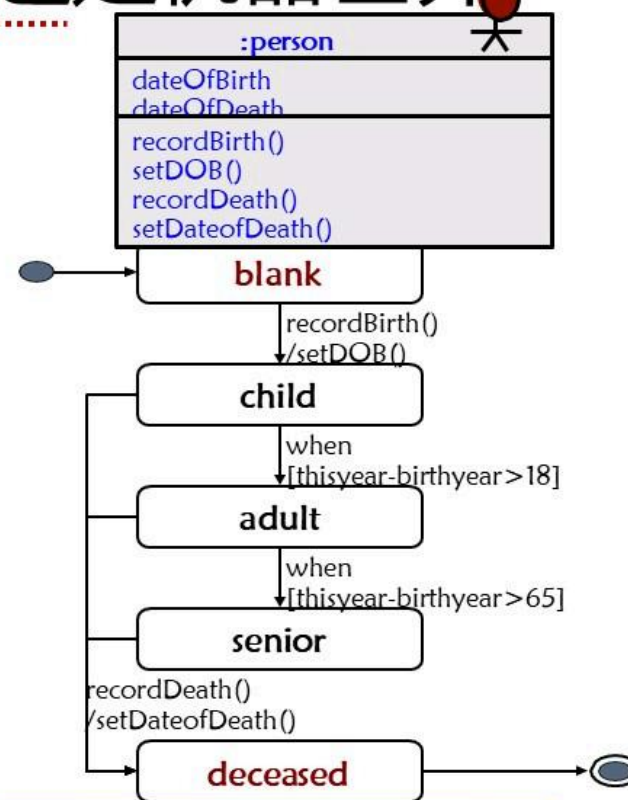
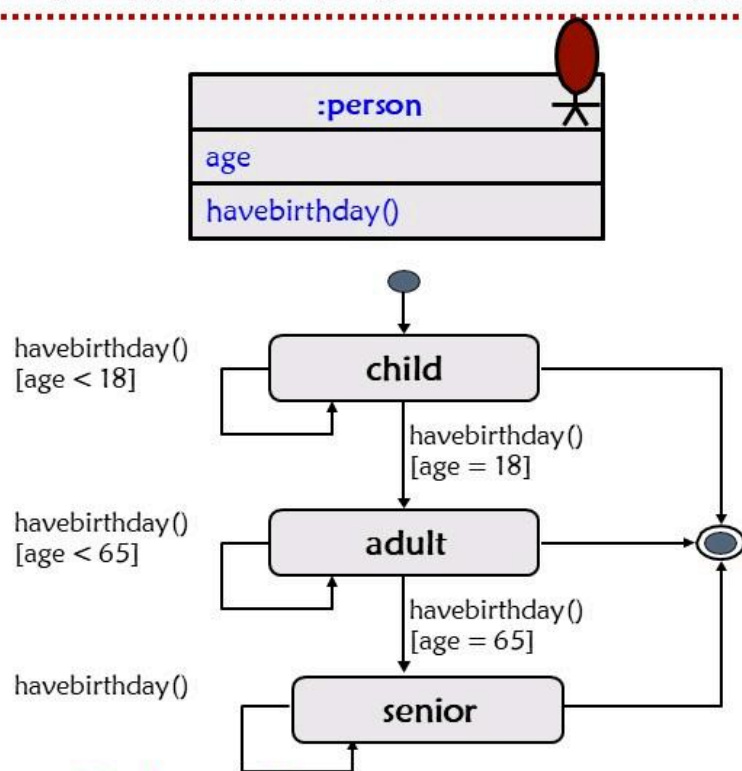
- 但往往状态空间中的局部更有探究的价值
 - 有一些状态是不可能出现的状态
 - 整数或实数值属性往往只在一定范围内取值
 - 通常, 我们只关注特定约束下的对象及其行为 例
如. 对于年龄, 我们经常选择以下的范围:
 - $\text{age} < 18; 18 \leq \text{age} \leq 65; \text{age} > 65$
 - 例如. 对于费用信息, 我们更关注的约束划分为:
 - $\text{cost} \leq \text{budget}, \text{cost} = 0, \text{cost} > \text{budget}, \text{cost} > (\text{budget} + 10\%)$

模型建立的过程——状态空间的分解

- 抽象之后的模型可以表达更多的状态序列

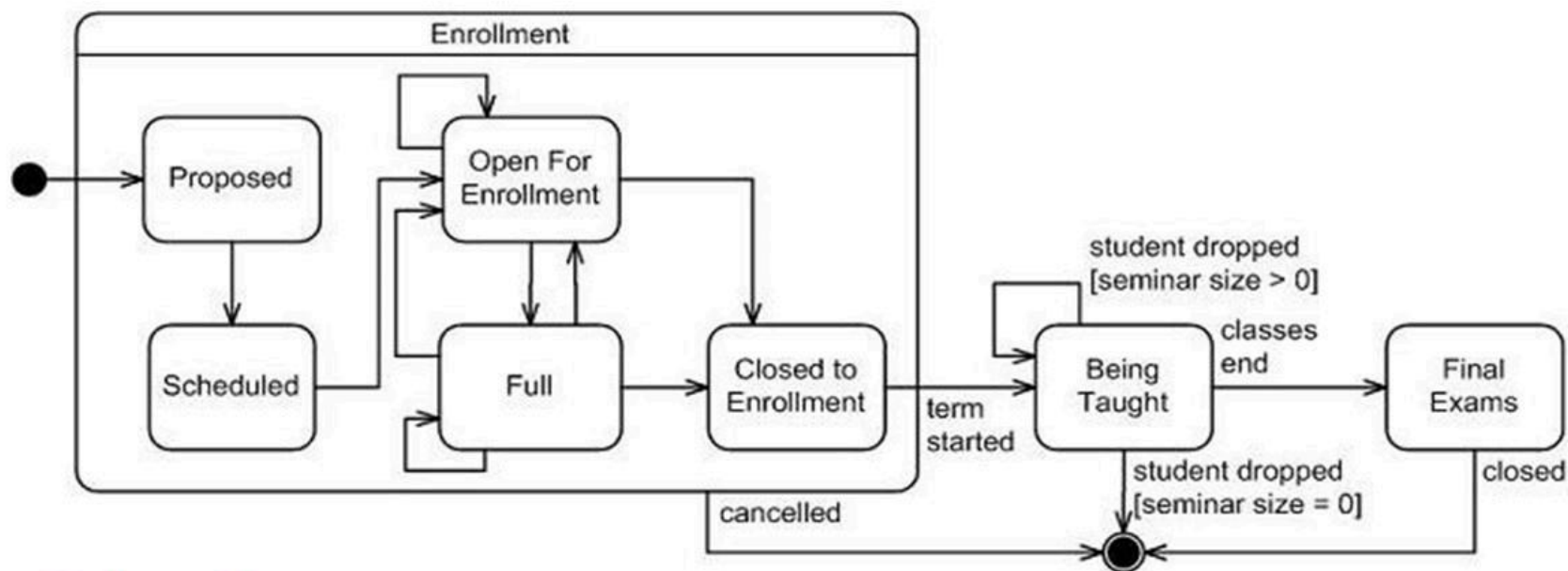


建模什么? ——现实世界还是机器世界



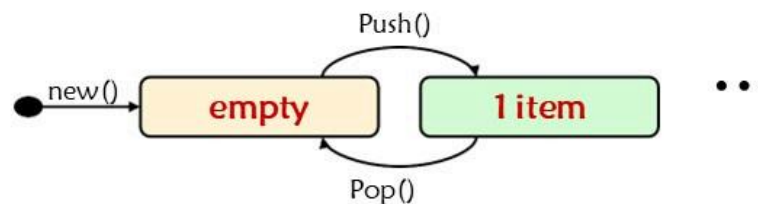
状态图 (State Chart/State Machine Diagram)

- 状态图用来表示一个类对象的全生命周期过程



状态图建模

- 建模元素
 - 状态
 - 事件
 - 状态转移
- 特殊的状态
 - 初始状态、结束状态
 - 组合状态、嵌套状态
 - 历史状态
- 状态图的绘制



状态的定义

一个对象生命期的一个阶段，该阶段中对象要满足一些特定的条件、执行特定的活动或等待某个（些）事件的发生。

- 体现为对象属性的取值
- 包含状态入口或出口行为描述
- 从不同的抽象层次分析对象，因此其状态是可嵌套（组合）的
- 在给定的场景下，对象状态是确定的，可满足或不满足某个状态

事件的定义

可以触发对象状态改变的外部刺激，也就是消息的发出与接收。

- 决定状态迁移何时发生



状态迁移

定义：

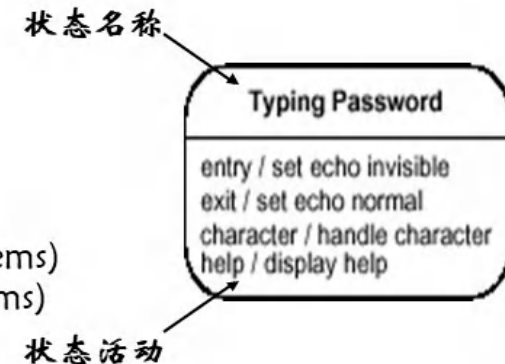
是状态之间的关系，当发生一个事件，条件满足时就会发生从源状态(Source State)到目标状态的转变(Object State)

- 当且仅当迁移条件满足时才能触发状态迁移
- 每个状态迁移都对应一个触发“事件”
- 同时还需要满足一定的“警戒条件(Guard Condition)”
- 当触发事件发生，或相关警戒条件满足时，进行相应的状态迁移
- 状态迁移的过程会伴随相关的对象操作



UML状态图中的状态(State)

- 一个状态表示在某个时间段内
 - 某个陈述是正确的
 - 例如. $(\text{budget} - \text{expenses}) > 0$
 - 某个动作正在执行或者在某个事件等待触发
 - 例如. 检查订单商品的存货(checking inventory for order items)
 - 例如. 等待缺货产品到货(waiting for arrival of a missing items)
- 状态相关的活动类型
 - **do/activity**
 - 只要处于这个状态, 某个活动就会一直执行, 直到离开这个状态
 - **entry/action** and **exit/action**
 - 当进入 (/离开) 某个状态时执行的动作
 - **include/stateDiagramName**
 - 调用另一个状态图, 形成嵌套的状态图



初始状态



结束状态

UML状态图中的迁移(Transitions)

迁移包括五部分:

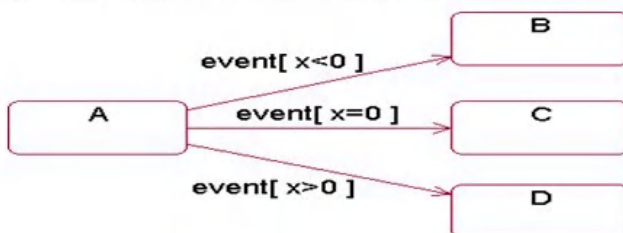
- 源状态(source state)、触发事件 (event trigger), 警戒条件(guard condition), 动作(action), 目标状态(target state).

源状态

事件名['(' 用逗号分隔的参数表 ') '][警戒条件] '/' 动作表达式

目标状态

- 对于给定的状态, 最终只能产生一个迁移, 因此从相同的状态出来的、事件相同的几个迁移之间的条件应该是互斥的。



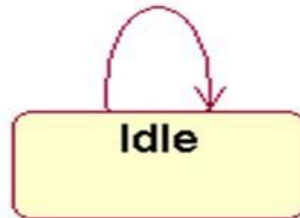
UML状态图中的事件(Event)

- 事件(Events)的意义在于系统需要了解正在发生什么
 - 状态图中, 事件仅需和系统或当前建模的对象相关
 - 从系统角度出发, 事件必须建模成一个瞬间可完成的动作
 - 例如. 完成工作, 考试未通过, 系统崩溃
 - 在OOD(面向对象设计)中通过传递消息的方式实现事件
- 在UML中, 有四种类型的事件
 - 变更事件(Change events), 当给定条件成立时就会发生状态变更事件
 - 调用事件(Call events), 当给定对象的操作被调用执行时会发生调用事件
 - 时间事件(Elapsed-time events), 表明时间段过去, 或某个特殊时间点的触发
 - 信号事件(Signal events), 当给定对象收到某实时信号

UML状态图中的事件(Event)——变更(Change)事件

- **变更事件(Change event):** 通过布尔表达式中变量的改变, 使得表示式成立的事件, 通过“when”关键字进行提示
- 例:

`when(temperature > 120) / alarm()`



- 变更事件和警戒条件(guard condition)的区别:
 - 警戒条件只在所相关的事件出现后计算一次, 如果值为false, 则不进行状态转移

UML状态图中的事件(Event)——调用(Call)事件

- **调用事件(Call event)**: 在这一类事件中, 状态迁移的动作会调用对象的方法
- 语法格式如下: **事件名 ([逗号分隔的参数列表])**
 - 其中参数列表中的参数格式为: **参数名: 类型**

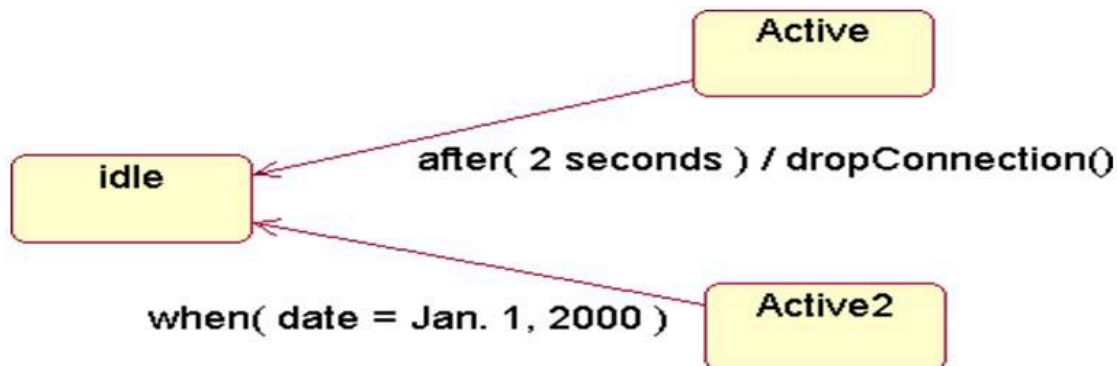
例:



UML状态图中的事件(Event)——时间(Time)事件

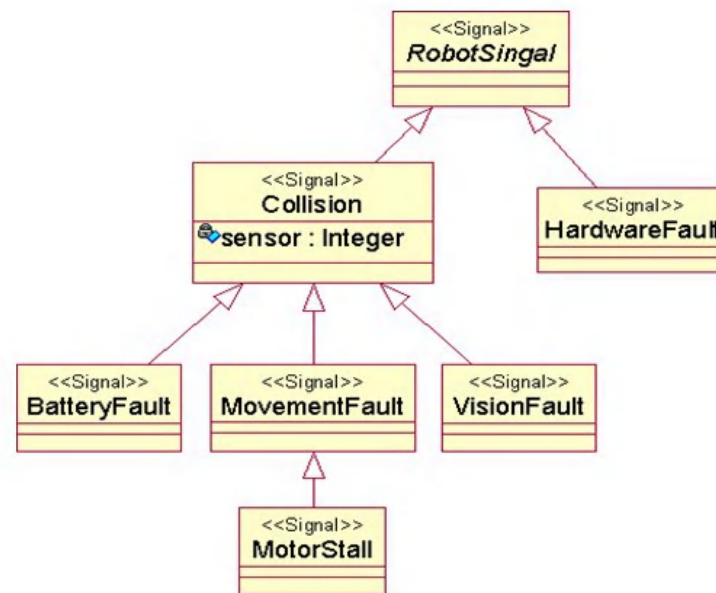
- **时间事件(Time event):** 通过时间表达式是否满足来表示事件, 例如一个绝对时间点的到来, 或者经过时间段过去后对象进入一个新状态
- 用关键字after或when表示

例:



UML状态图中的事件(Event)——信号(Signal)事件

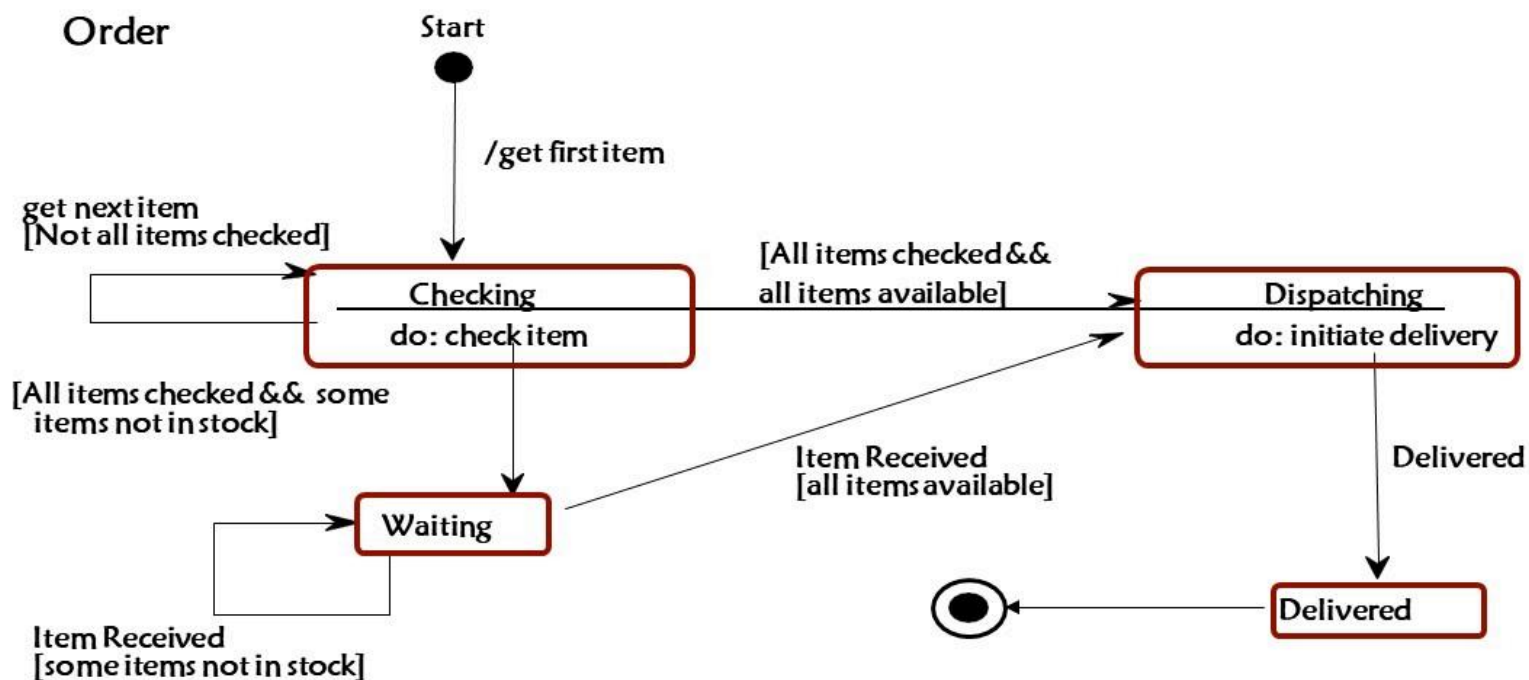
- **信号事件(Signal event):** 表示接受一个对象发送的信号（信息）的事件，有可能引发状态迁移（状态改变）
- 语法格式如下：
事件名（[逗号分隔的参数列表]）
- 信号事件与调用事件的区别：
 - 信号事件是一个异步事件，调用事件一般是一个同步事件。



UML状态图中的动作(Action)

- **动作**是在状态内部或者状态间迁移时执行的原子操作
- 两种特殊的动作:入口动作(entry action)和出口动作(exit action)
 - Entry动作: 进入状态时执行的活动, 格式如下:
 'entry' '/' action-expression
 - Exit动作: 退出状态时执行的活动, 格式如下:
 'exit' '/' action-expression(其中 action-expression 可以引用对象本身的属性和输入事件的参数)

例：订单处理

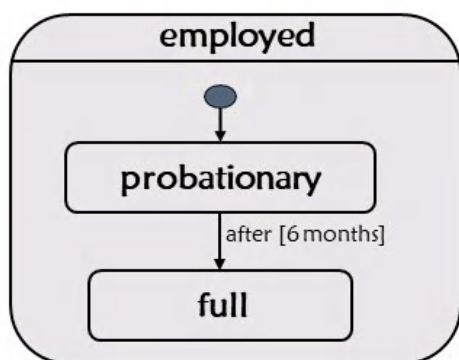


UML状态图中的组合状态 (Superstates)

- 可以通过状态嵌套的方式简化图表
 - 一个组合状态可以包含一个或多个状态
 - 组合状态可以实现从不同抽象层次去体现状态图

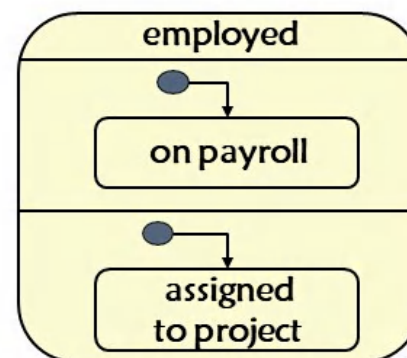
“OR” 的组合状态

- 处于组合状态时只能满足其中一个子状态

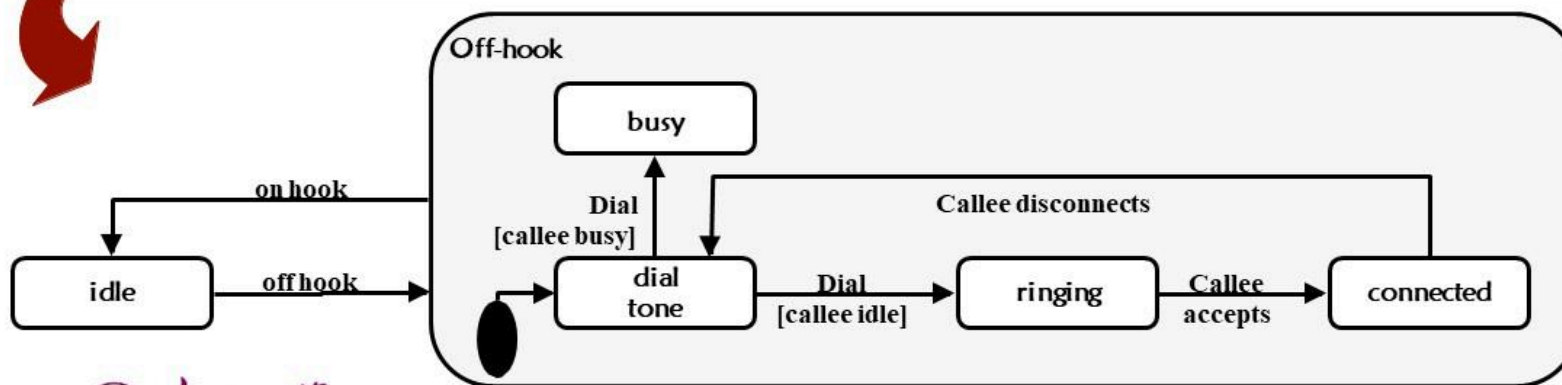
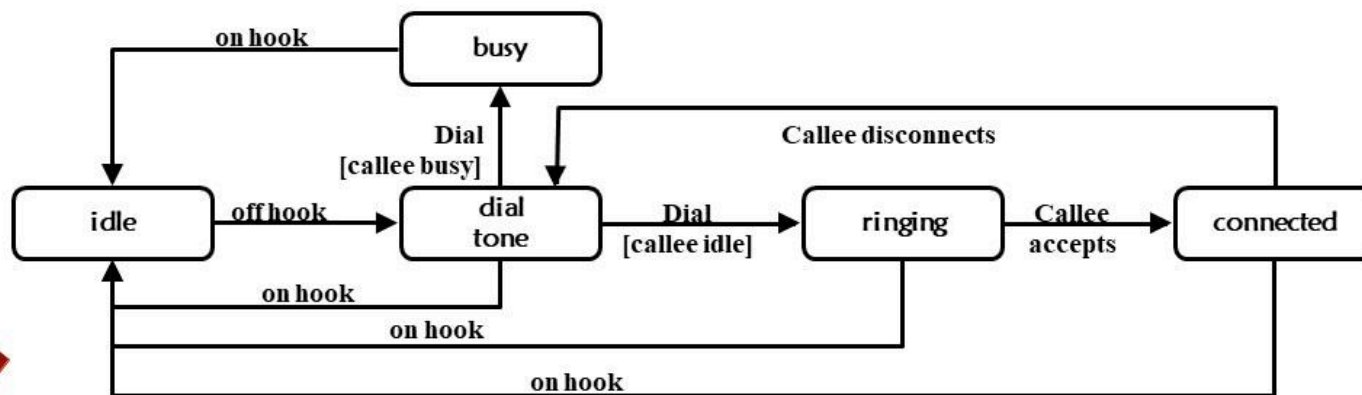


“AND” 的组合状态(并发状态)

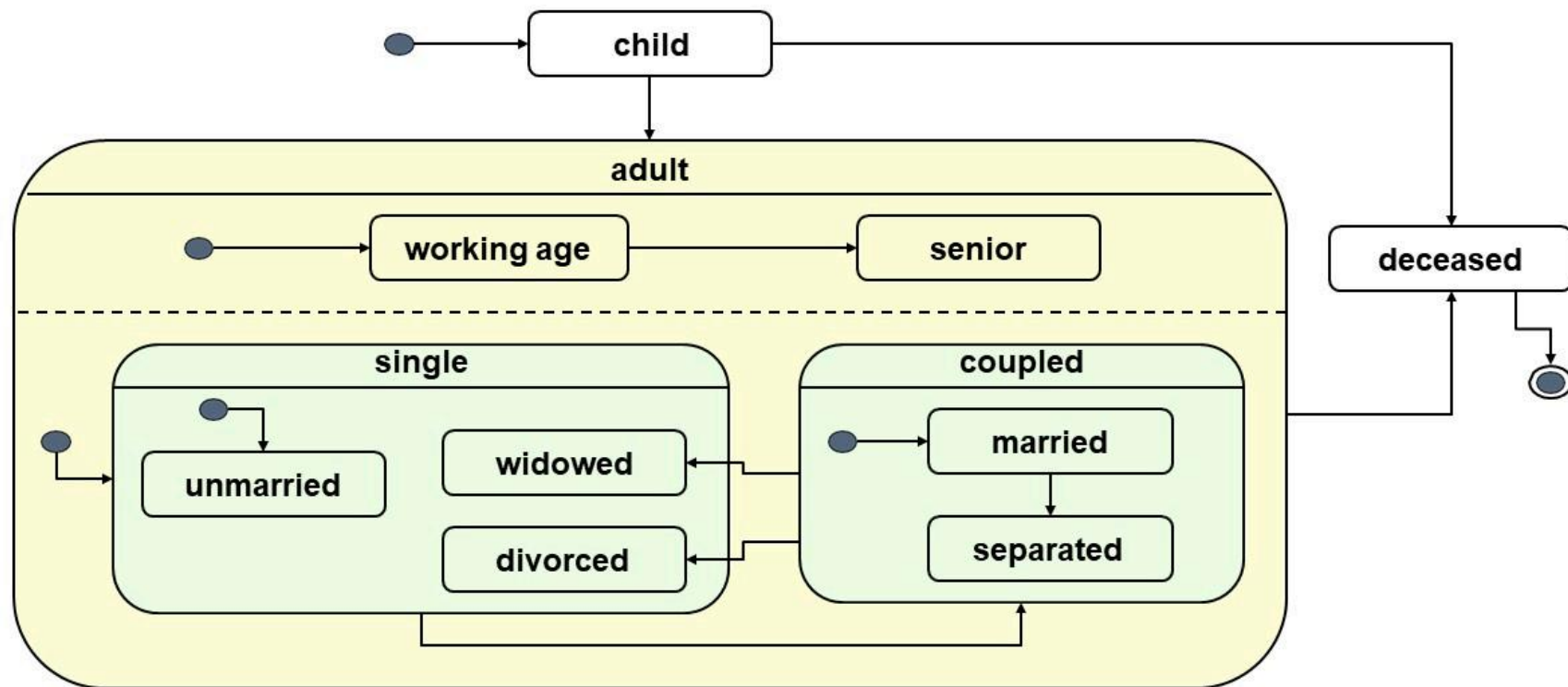
- 处于组合状态时，满足所有的子状态
- 通常，AND的子状态会进一步嵌套为OR的子状态



组合状态的例子

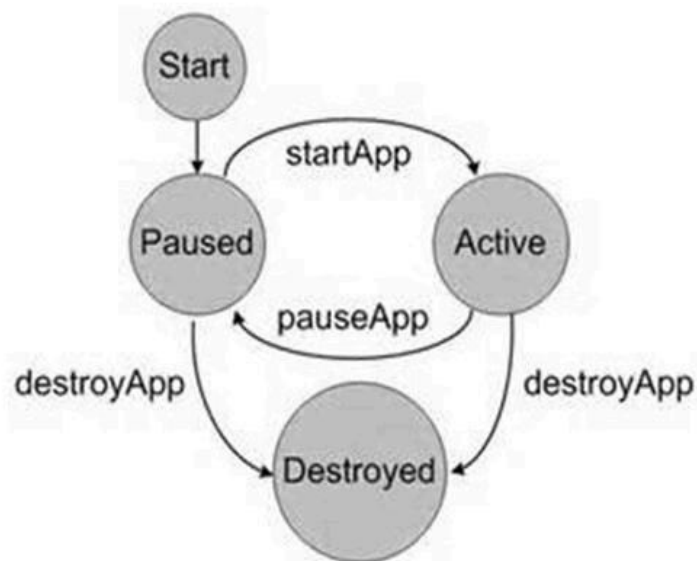


组合状态的例子

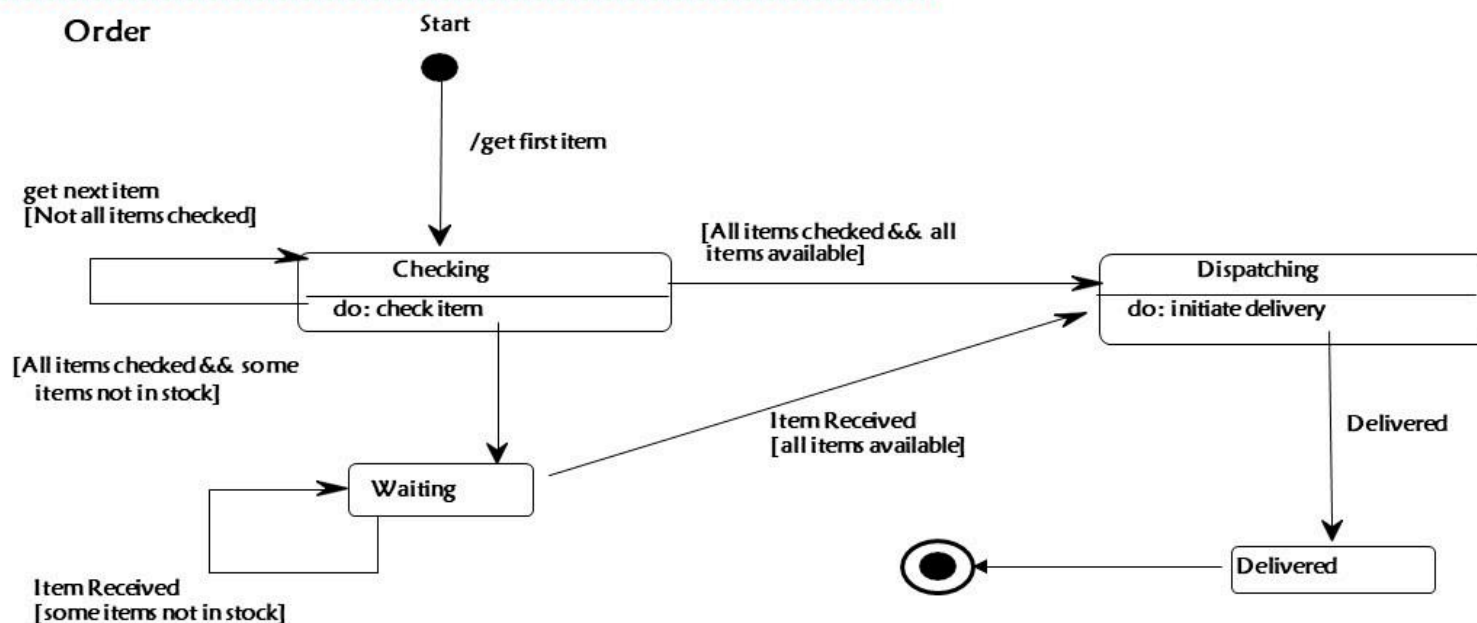


组合状态的状态迁移

- 指向组合状态边界的状态迁移等价于指向该组合状态初态的迁移
 - 所有属于该组合状态的入口条件将被执行
- 从组合状态边界转出的迁移等价于从该组合状态的终态发出迁移
 - 所有出口条件均将被执行
- 迁移可直接指向组合状态的子状态



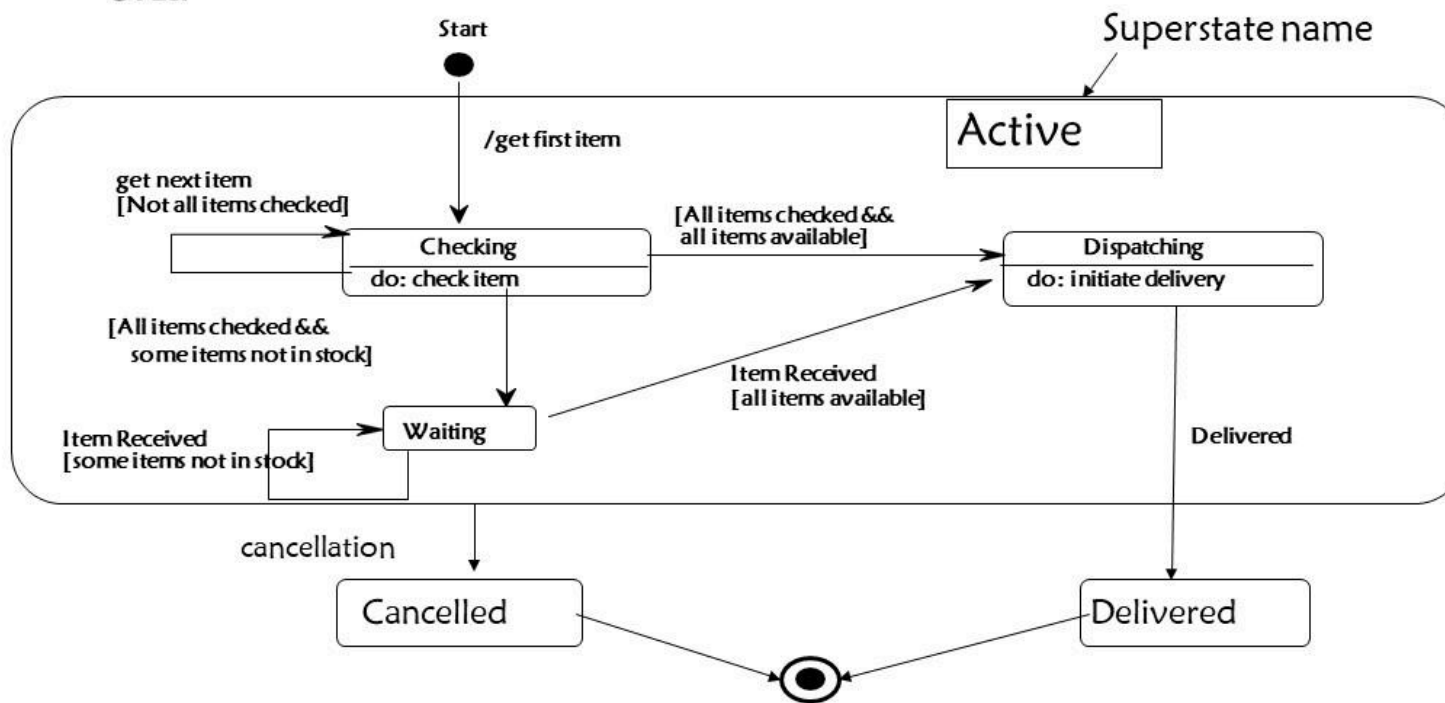
练习：组合状态



练习：在图中增加一个新的状态和相关的状态迁移，表示在物品投递之前的任何环节都可以取消订单

参考答案

Order



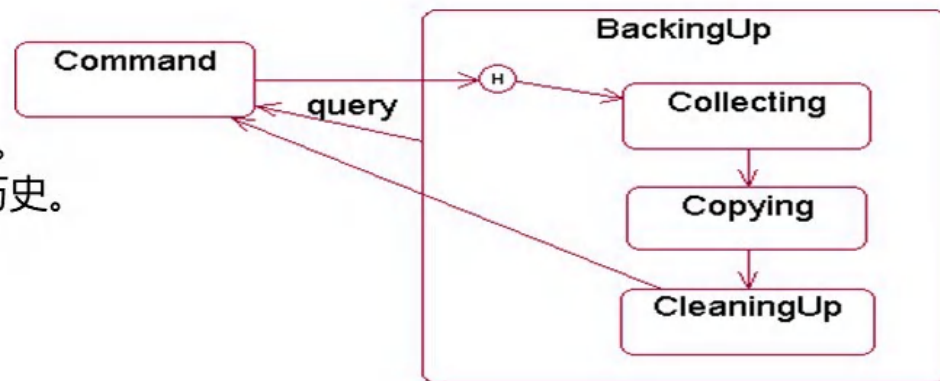
UML状态图中的历史状态(History State)

- 历史状态是一种伪状态。当激活这个状态时，会保存从组合状态中退出时所处的子状态，用H表示
- 当再次进入组合状态时，可直接进入到这个子状态，而不是再次从组合状态的初态开始。

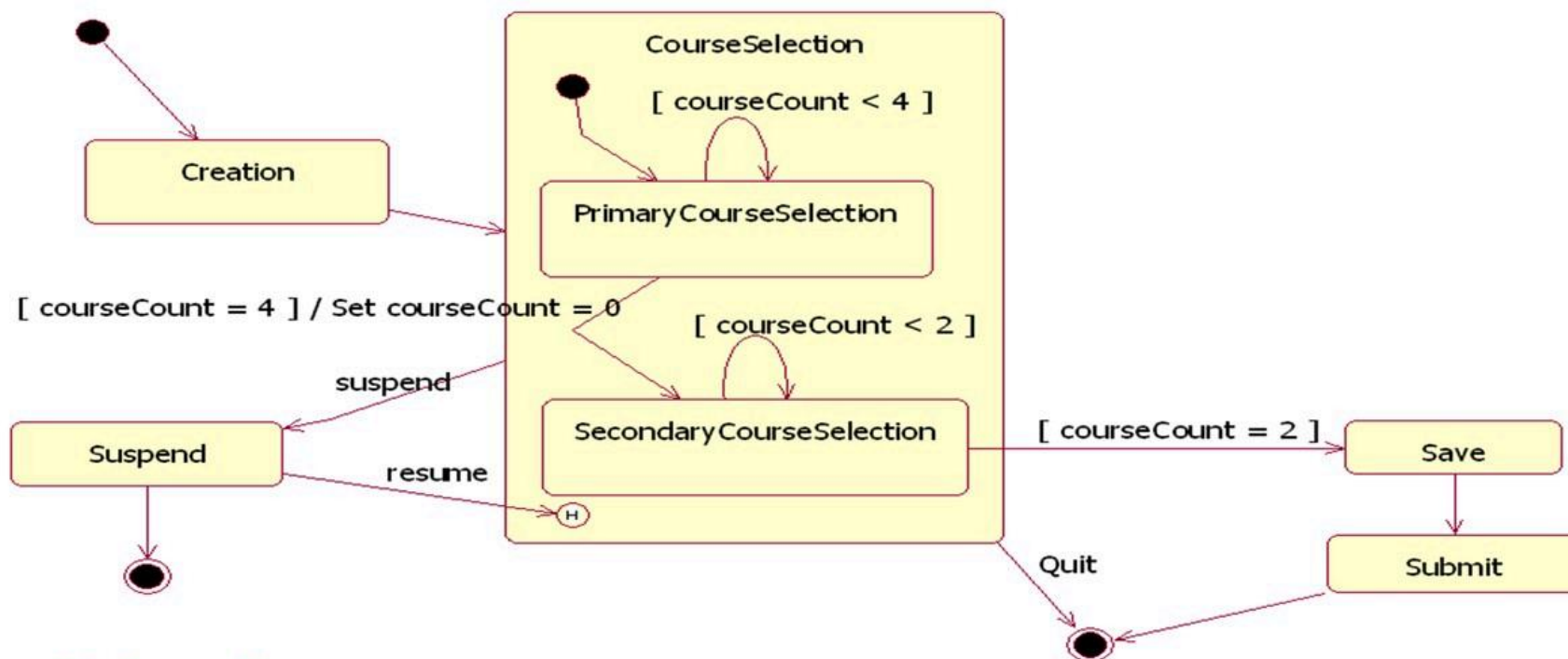
- H和H*的区别：

- H只记住最外层的组合状态的历史。
- H*可记住任何深度的组合状态的历史。

例：历史状态的例子。

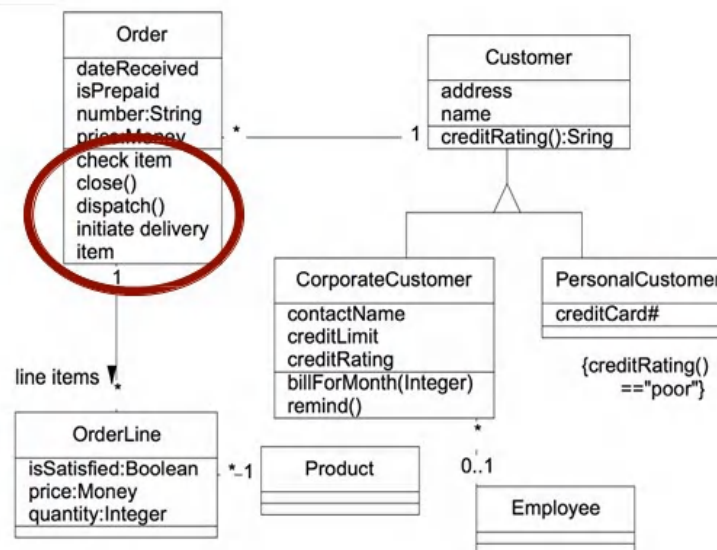
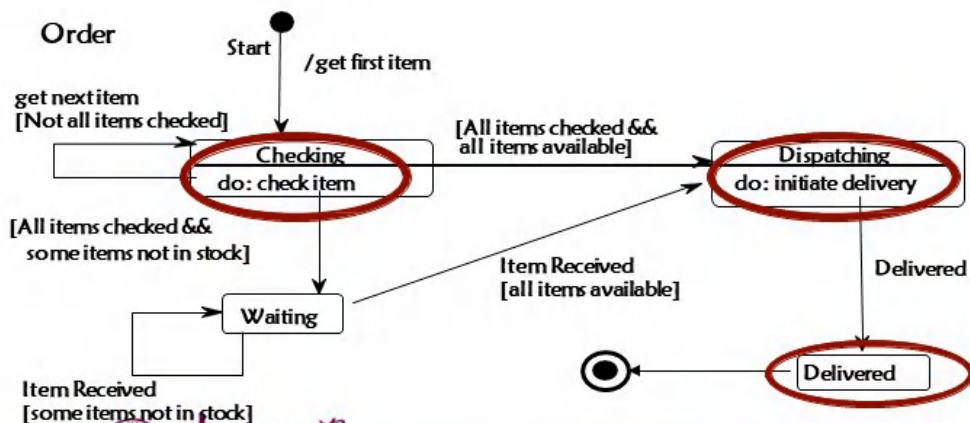


例：课程注册



状态图与其他UML图的关系

- 状态图应针对类图中具有重要行为的类进行建模
- 状态图中的操作定义等价于类图中的操作定义
- 状态图中的动作对应于类中的一个具体操作
- 事件为顺序图中该对象的输入消息



状态图建模风格

- 建模风格1：把初态放置在左上角；把终态放置在右下角
- 建模风格2：用过去式命名转移事件
- 建模风格3：警戒条件不要重叠
- 建模风格4：不要把警戒条件置于初始转移上

单选题 1分

下面关于对象状态建模的说法中，正确的是：

- ☐ A 大部分对象的状态空间都是有限的
- ☐ B 对象状态建模要穷举对象能够到达的所有状态，保证完整性
- ☒ C 大部分对象的状态空间大小是由它的属性取值决定的
- ☐ D 对象的状态数量是由它对外提供的操作的数量决定的

单选题 1分

状态迁移的发生不会受到哪个因素的影响？

- ☒ A 目标状态
- ☐ B 外部事件
- ☐ C 警戒条件
- ☐ D 迁移动作

单选题 1分

下面关于UML状态图的说法正确的是：

- ☒ A UML状态图中的状态可以分解为“与”状态，以及“或”状态，但是都可以转化为基本状态机来表示
- ☐ B UML状态图中的状态是原子的，不可再分
- ☐ C UML状态图中的状态只可以分解为“与”状态，表示可以两种状态并存
- ☐ D UML状态图中的状态只可以分解为“或”状态，表示只能选择其中之一

单选题 1分

关于状态图与其他UML图的关系，说法不正确的是

- ☐ A 状态图中的事件为顺序图中该对象的输入消息
- ☐ B 状态图是对一个关键类对象的状态进行建模
- ☐ C 状态图中的动作定义等价于类图中的操作定义
- ☒ D 状态图应针对类图中所有的类进行建模

单选题 1分

以下状态迁移上的警戒条件定义中，哪一组是最合适的？

- ☒ A $x > 0, x = 0, x < 0$
- ☐ B $x \geq 0, x \leq 0$
- ☐ C $x > 0, x < 0$
- ☐ D $x > 0, x = 0$

单选题 1分

对象的状态建模，其前提条件是？

- ☐ A 对象的属性取值是有限个数的
- ☒ B 所有的对象都是有状态的
- ☐ C 具有类图建模
- ☐ D 对象是现实世界存在的

多选题 1分

以下说法哪个是正确的

- ☒ A 当前对象接收到的事件是由当前对象对外提供的某种服务
- ☐ B 当前对象接收到的事件是由外部对象对外提供的某种服务
- ☒ C 当前对象所发出的动作是由外部对象提供的某种服务
- ☐ D 当前对象所发出的动作是由当前对象提供的某种服务