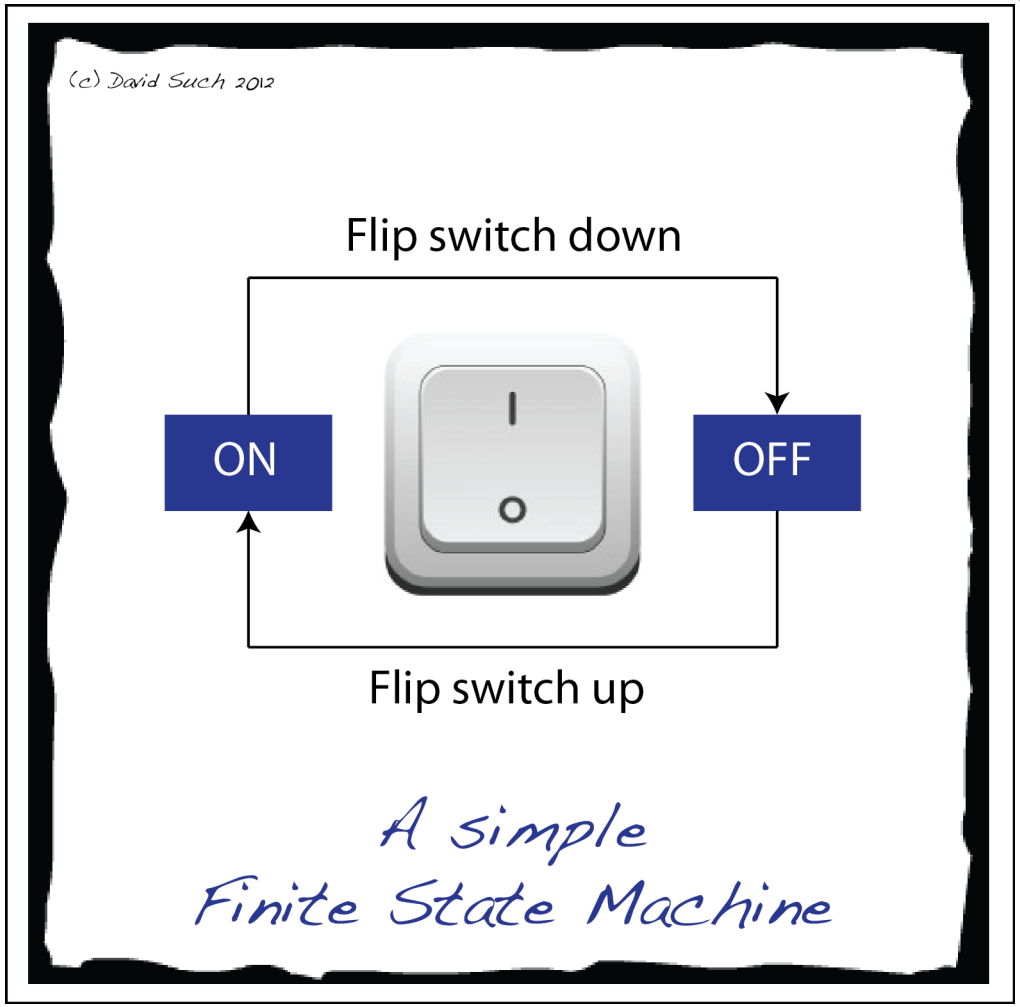
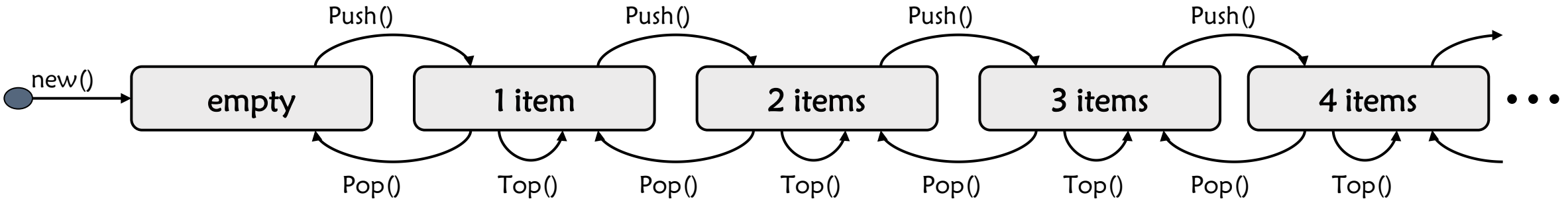
“状态”建模

- 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



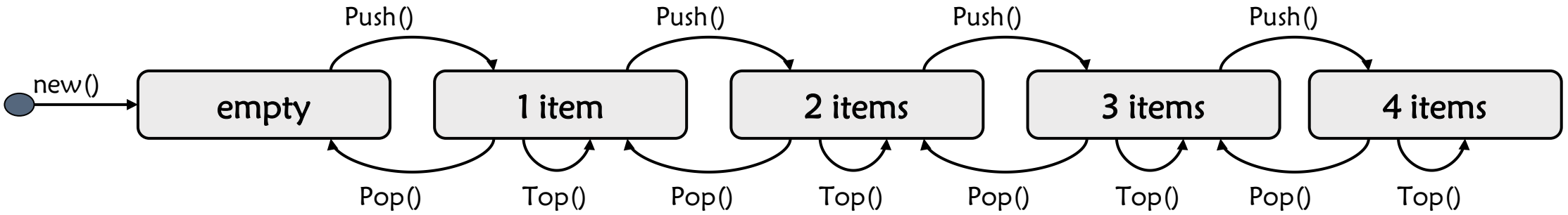
对象及其状态

- 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



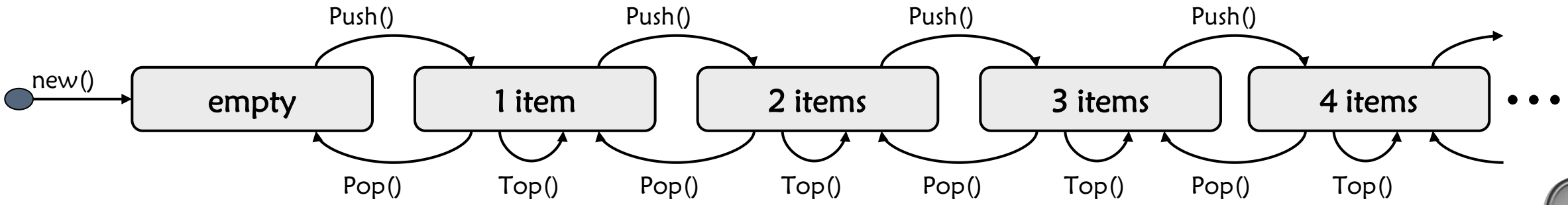
有限状态机

- 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 $(maxint)^5+1$ 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：

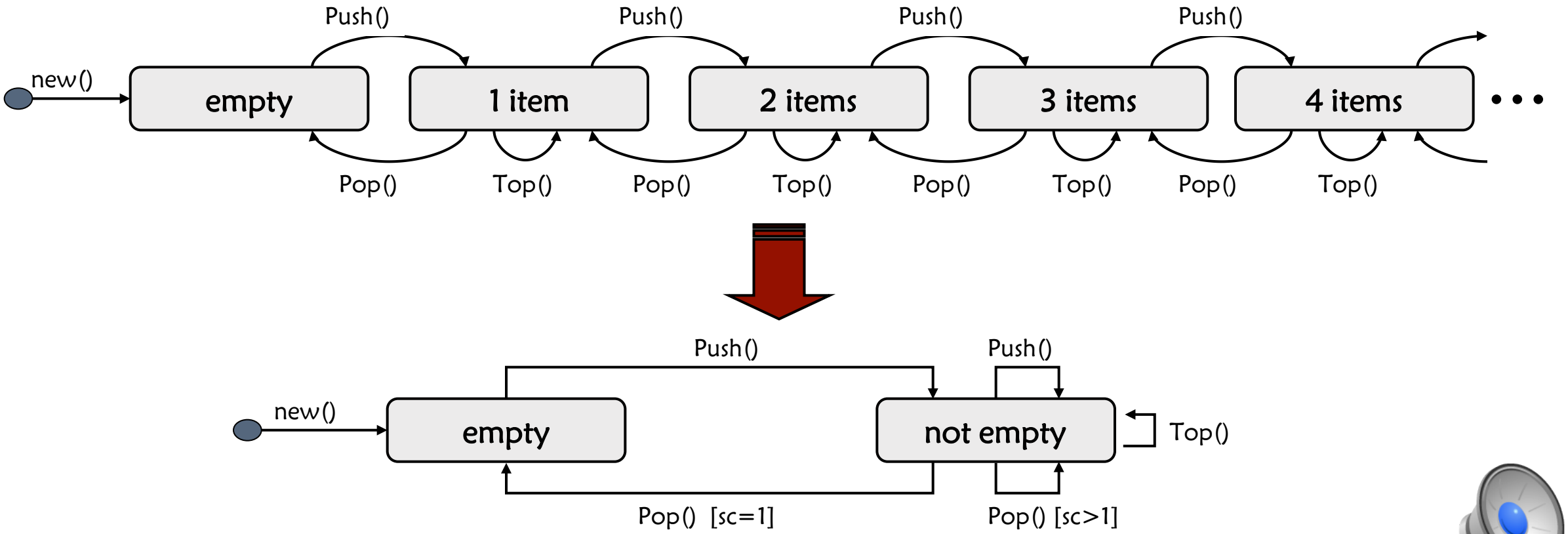
    $age < 18; 18 \leqslant age \leqslant 65; age > 65$

    例如. 对于费用信息,我们更关注的约束划分为：

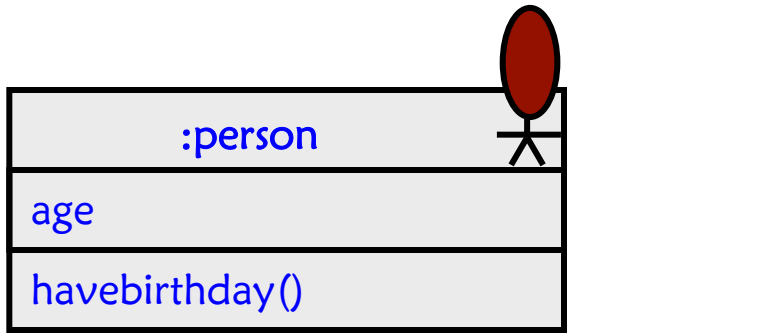
    $cost \leqslant budget, cost = 0, cost > budget, \ cost > (budget + 10\%)$

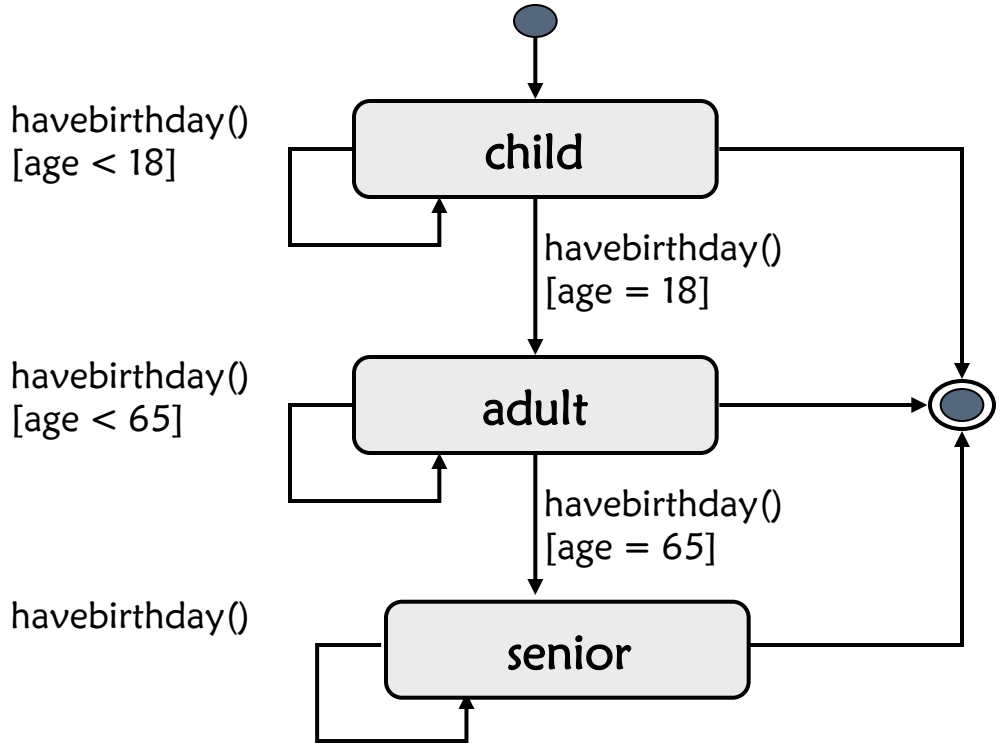
模型建立的过程——状态空间的分解

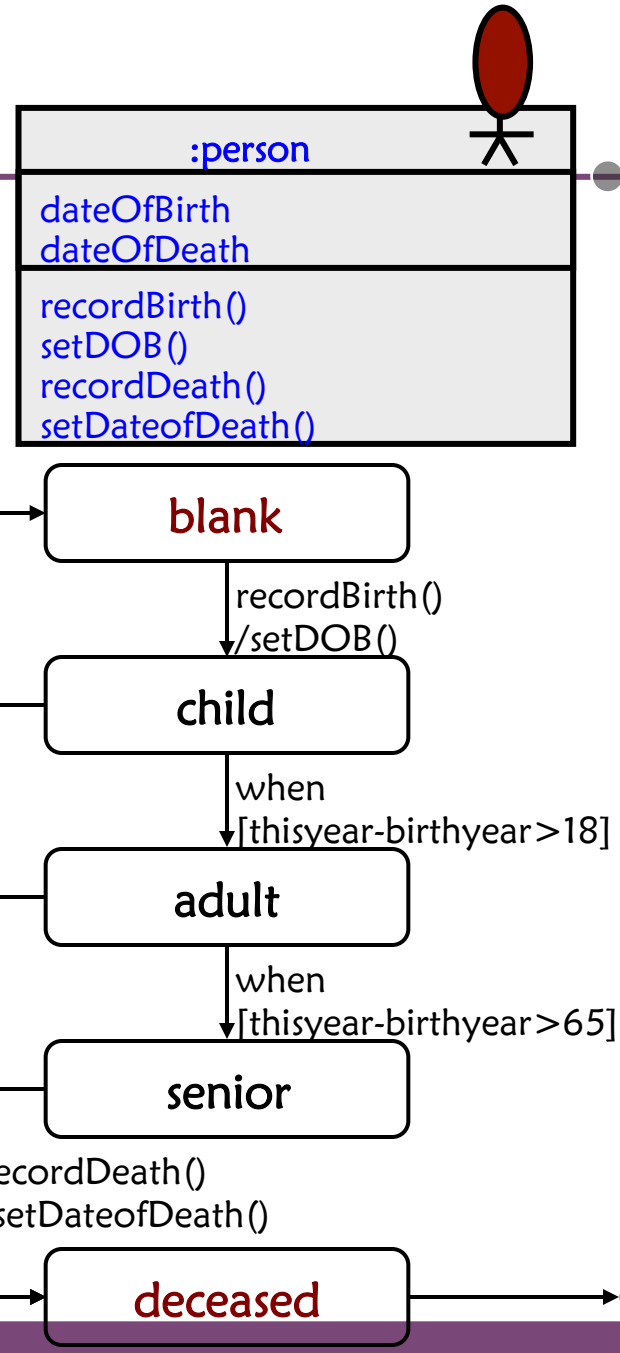
- 抽象之后的模型可以表达更多的状态序列
  - 例如. 上面的模型并不能防止pop()多于push()的序列出现
  - 仍然表达了很多信息



建模什么？——现实世界还是机器世界







对什么建模?

应用领域

机器领域

(D) 应用领域实体的可观测状态

- 例如. 一个电话的状态有空闲，响铃，接通……
- 模型表达了实体可能处于的状态，以及什么操作会造成状态的变化
- 是一种描述(Indicative)模型：描述实体当前状态

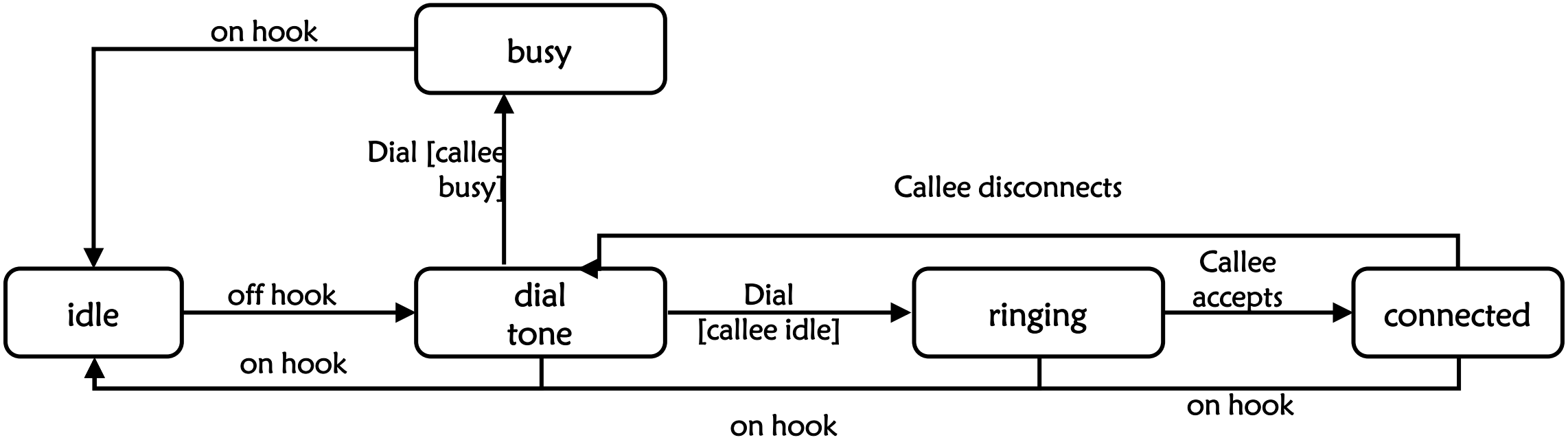
(R) 应用领域实体所需要的行为

- 例如. 电话交换机当且仅当被呼叫的人接受了电话请求才会连接两个电话
- 模型可以区分一系列状态序列或操作路径是否能达到预期的结果
- 是一种愿望(Optative)模型：描述动作及其预期结果

(S) 机器领域实体的具体行为

- 例如，当用户摁“连接”摁扭，打入的电话将会被接通
- 模型表达了机器该如何响应输入的事件
- 这是一个愿望(Optative)模型，所有的事件是共有的。

思考题



这是描述（indicative）模型还是愿望（optative）模型呢？