

软件工程

第10章 需求建模 ——行为和模式

徐本柱 软件学院

2018-09

主要内容

- ❖ 生成行为模型
- ❖ 识别用例事件
- ❖ 状态表达
- ❖ 需求建模的模式

引言

- ❖ 基于场景建模和建立基于类模型足够了吗？
- ❖ “看情况而定”
- ❖ 有些类型软件，用例是唯一可行的需求建模方法
- ❖ 其他类型，需要OO方法，开发基于类的模型
- ❖ 另外情形，要对复杂应用软件需求进行检测：
 - 数据在系统中移动时如何转换？
 - 外部事件如何处理？
 - 。 。 。 。 。 。

10.1 生成行为模型

- ❖ **行为模型**显示了软件对外部事件或激励作出响应
- ❖ 生成模型按如下步骤进行：
 - ❖ 1 评估所有用例，以保证完全理解系统内的交互顺序
 - ❖ 2 识别驱动交互顺序的事件，并理解这些事件如何与特定的对象相互关联
 - ❖ 3 为每个用例生成序列
 - ❖ 4 创建系统状态图
 - ❖ 5 评审行为模型以验证准确性和一致性

10.2 识别用例事件

- ❖ 用例表现了涉及的参与者和系统的活动顺序
 - ❖ 只要系统和参与者之间交换了信息就发生事件。
- ❖ 事件
 - ❖ 不是被交换的信息
 - ❖ 而是已交换信息的事实
- ❖ 一旦确定了所有的事件
 - ❖ 这些事件将被分配到所涉及的对象
 - ❖ 对象负责生成事件或识别已经在其他地方发生的事件

SafeHome实例[25]

房主使用键盘键入4位密码，该密码和保存在系统中的验证密码相比较。如果密码不正确，控制面板将鸣叫一声并复位以等待下一次输入；如果密码正确，控制面板等待进一步的操作。

用例场景中加下划线的部分表示事件。应确认每个事件的参与者，应标记交换的所有信息，而且应列出每个条件或限制。

举个典型事件的例子，考虑用例中加下划线的“房主使用键盘键入4位密码”。在分析模型的环境下，对象**Homeowner**²⁵向对象**ControlPanel**发送一个事件，这个事件可以称作“输入密码”。传输的信息是组成密码的4位数字，但这不是行为模型的本质部分。重要的是注意到某些事件对用例的控制流有明显的影响，而其他的事件对控制流没有直接的影响。例如，事件“输入密码”不会显式地改变用例的控制流，但是事件“比较密码”（从与事件“该密码和保存在系统中的验证密码相比较”的交互中得到）的结果将直接影响到SafeHome软件的信息流和控制流。

10.3 状态表达

- ❖ 在行为建模中，必须考虑两种不同的状态描述：
 - ❖ (1) 系统执行其功能时每个类的状态
 - ❖ (2) 系统执行其功能时从外部观察到的系统状态
- ❖ 类状态具有被动和主动两种特征
 - ❖ 被动状态：某个对象所有属性的当前状态
 - ❖ 主动状态：指对象进行持续变换和处理时的当前状态
 - ❖ 事件（触发器）才能迫使对象从一个主动状态转移到另一个主动状态

分析类的状态图

- ❖ UML 状态图是一种行为模型
 - ❖ 为每个类表现活动状态和
 - ❖ 导致这些活动状态变化的事件（触发器）
- ❖ 图10-1说明了SafeHome安全功能中ControlPanel类的状态图。

分析类的状态图

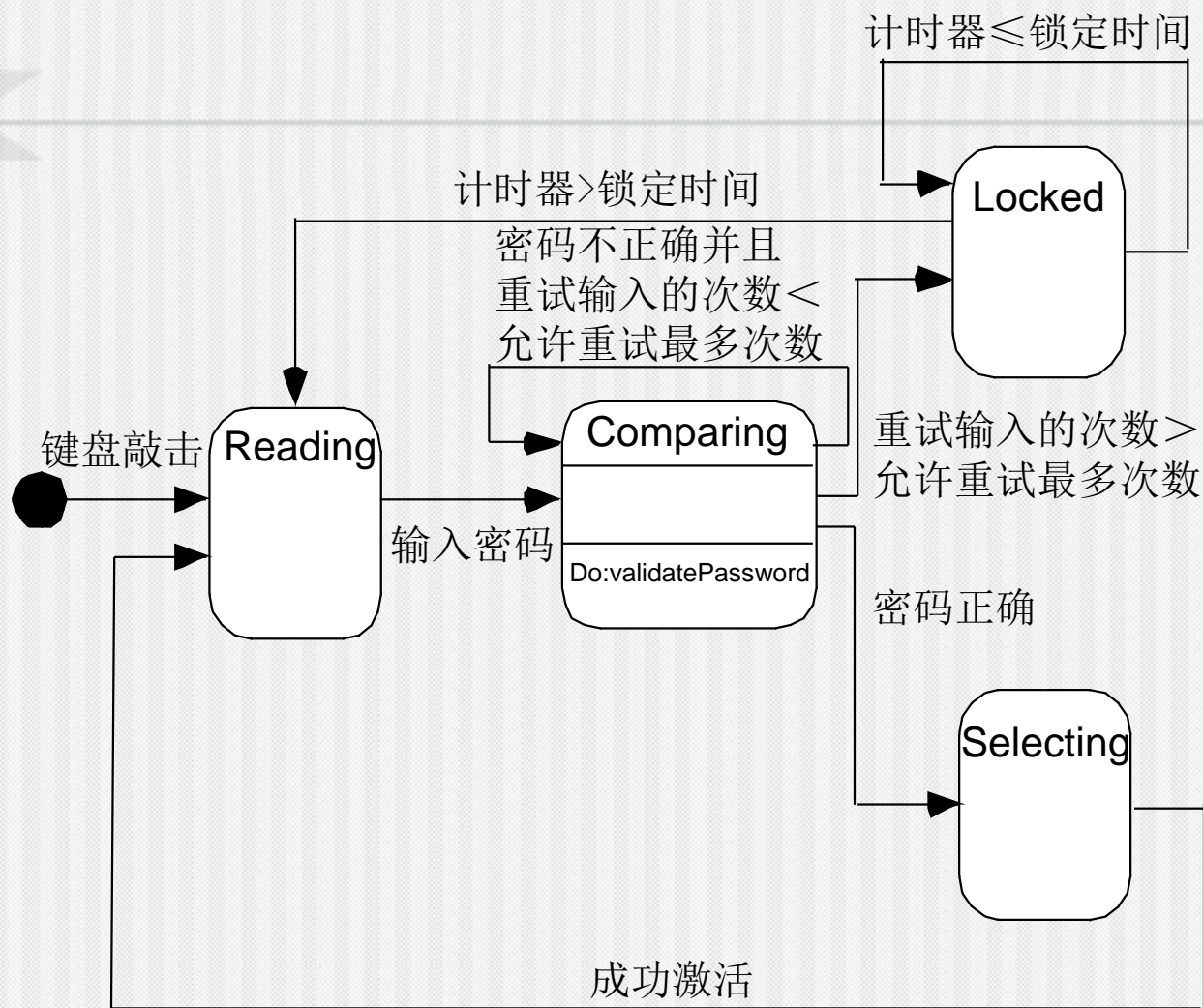


图10-1 ControlPanel类的状态图

分析类的状态图

❖ 还可以说明 **守卫** 和 **动作**

❖ **守卫** 是为了保证转移发生而必须满足的一个布尔条件

❖ 通常依赖于某个对象的一个或多个属性值

❖ 即保护依赖于对象的被动状态

❖ **动作** 是与状态转移同时发生或作为状态转移的结果

❖ 通常动作包含对象的一个或多个 **操作**

顺序图—用例的速记版

- ❖ 第二种表现行为的方式在UML中称作**顺序图**
- ❖ 用时间函数表现如何引发事件从一个对象流到另一个对象
 - ❖ 表明事件如何引发从一个对象到另一个对象的转移。
 - ❖ 一旦通过检查用例确认了事件
 - ❖ 建模人员就创建了一个**顺序图**
 - ❖ 表现了导致行为从一个类到另一个类的关键类和事件
- ❖ 图10-2给出了SafeHome安全功能的部分顺序图
 - ❖ 每个箭头代表一个事件，及如何引导对象之间的行为
 - ❖ 时间纵向（向下）度量
 - ❖ 窄的纵向矩形表示处理某个活动的时间

顺序图

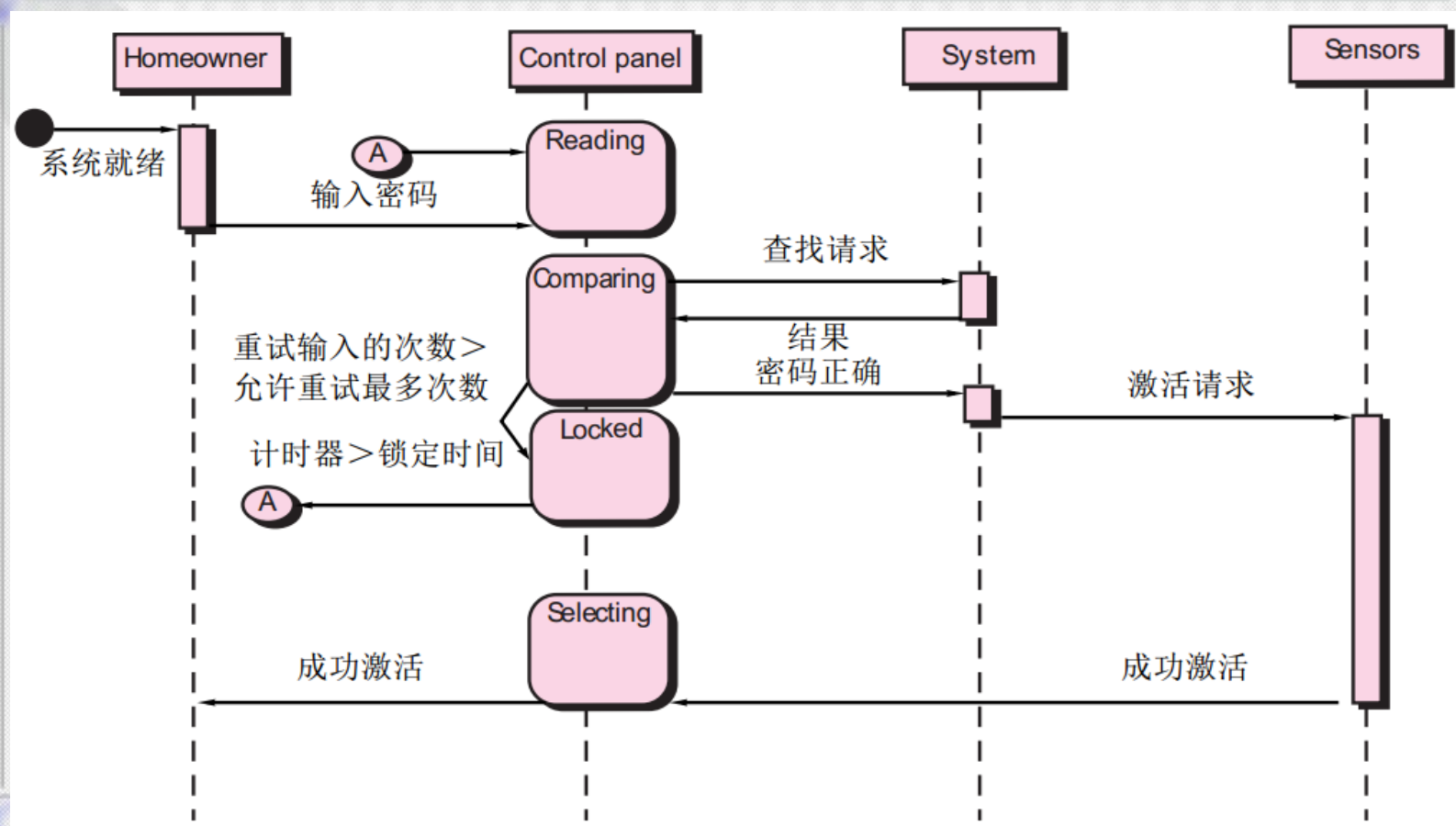


图10-2 SafeHome安全功能的顺序图(部分)

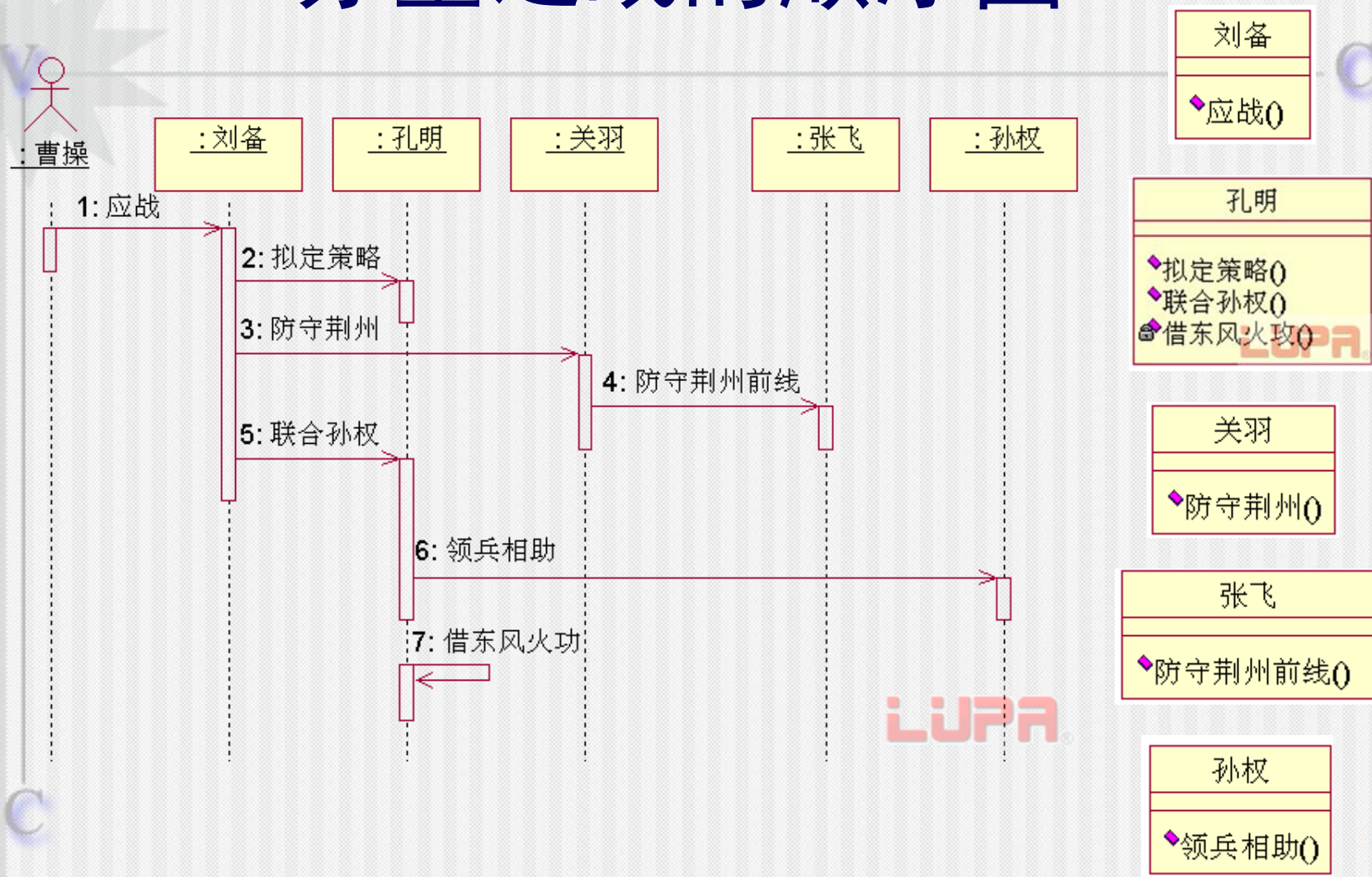
系统状态

- ❖ **状态**——一组可观察到的情况下,在给定的时间描述系统的行为
- ❖ **状态转移**——从一个状态转移到另一个状态
- ❖ **事件**——发生时系统将表现出某种可以测的行为
- ❖ **动作**——与状态转移同时发生的或者它作为状态转移的结果

行为建模

- ❖ 列出系统的不同状态（系统如何表现？）
- ❖ 指出系统如何从一种状态转移到另一种状态（系统如何改变状态？）
 - 指出事件
 - 指出动作
- ❖ 绘制状态图或顺序图

赤壁之战的顺序图



10.4 需求建模的模式

- ❖ **软件模式**是获取领域知识的一种机制
 - ❖ 遇到新问题时可以反复使用
 - ❖ 领域知识在同一应用领域中用于解决新问题
 - ❖ 通过模式获取的领域知识可模拟用于不同的应用领域
- ❖ 分析模式是在需求工程工作中发现了模式。一旦发现模式则记载“明确的常见问题：
 - ❖ 哪种模式适用
 - ❖ 规定的解决方案
 - ❖ 在实践中使用模式的假设和约束
 - ❖ 以及关于模式的某些常见的其他消息

- ❖ 在需求模型的描述中最基本的元素是用例
- ❖ 一套连贯用例可以成为服务于发现一个或多个分析模式的基础
- ❖ 语义分析模式 (**semantic analysis pattern, SAP**) “是描述了一小套连贯用例，这些用例一起描述了通用应用的基础” [Fer00]

例子

- ❖ 以下我们考虑要求控制和监控“实时查看摄像机”和汽车临近传感器的软件用例：

用例： *监控反向运动*

描述： 当车辆安装了反向齿轮，控制软件就能从后向视频摄像机将一段视频输入到仪表板显示器上。控制软件在仪表板显示器上叠加各种各样距离和方向的线，以便车辆向后运动时驾驶员能保持方向。控制软件还能监控临近传感器，以判定在车后方10英尺内是否有物体存在。如果临近传感器检测到某个物体在车后方 x 英尺(有车速决定)内就会让车自动停止。

例子

- ◆ 在需求收集和建模阶段，本用例包含（在一套连贯用例中）各种将要精炼和详细阐述的功能。
- ◆ 无论完成得如何精炼，建议用例要简单，但还要广泛地适用于SAP，即具有基于软件的监和在一个物理系统中对传感器和执行器的控制。
- ◆ 本例中，“传感器”提供临近信息和视频信息。
“执行器”用于车辆的停止系统（如果一个物体离车辆很近就会调用它）。
- ◆ 但是更常见的情况是发现大量的应用模式——**Actuator-Sensor**（执行器-传感器）。

执行器-传感器 模式—I

模式名：执行器-传感器

目的：详细说明在嵌入系统中的各种传感器和执行器。

动机：嵌入系统常有各种传感器和执行器。这些传感器和执行器都直接或间接连接到一个控制单元。虽然许多传感器和执行器看上去十分不同，但它们的行为是相似的，足以让它们构成一个模式。这个模式显示了如何为一个系统指定传感器和执行器，包括属性和操作。**执行器-传感器**模式为**被动传感器**使用“**拉机制**”（对消息的显示要求），为**主动传感器**使用“**推机制**”（消息广播）。

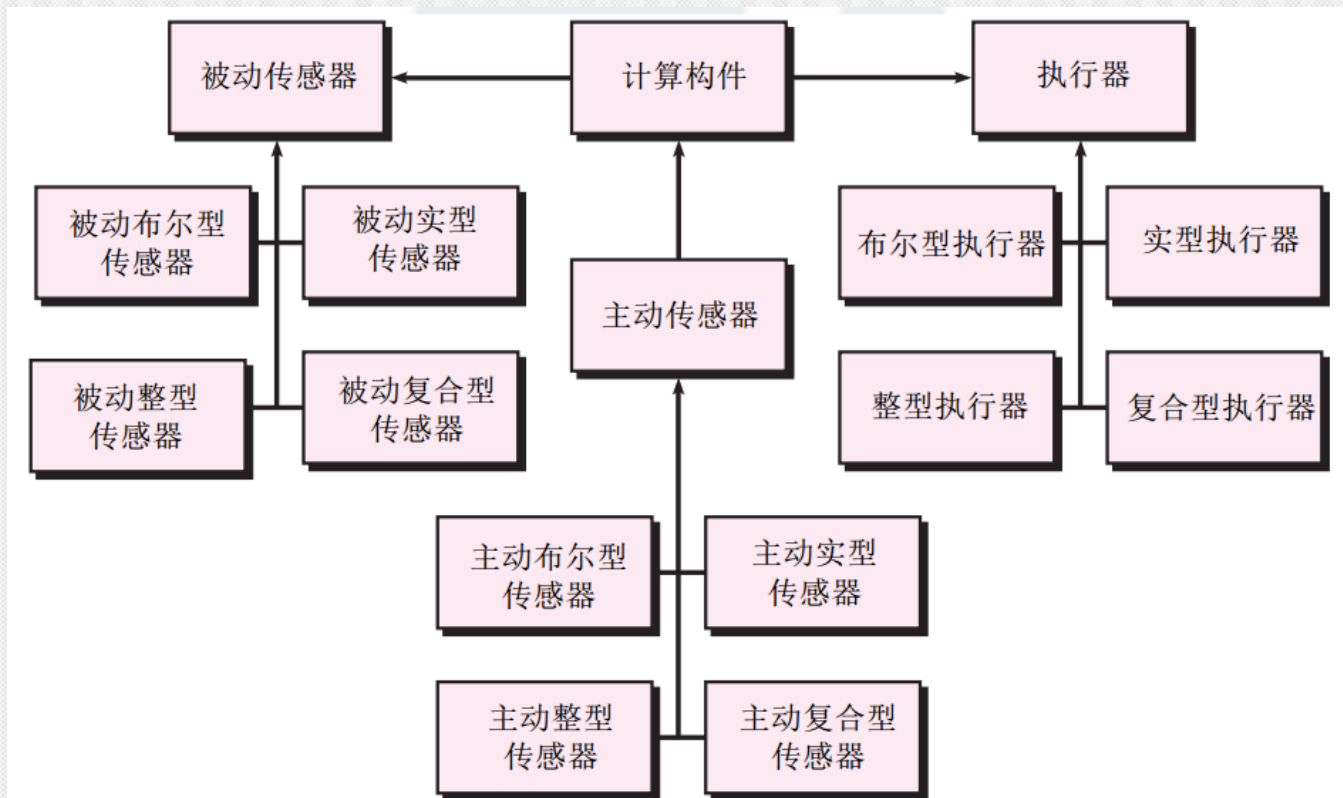
约束：

- 每个被动传感器必须有某种方法读取传感器的输入和表示传感器值的属性。
- 每个主动传感器必须能在其值发生变更时广播更新消息。
- 每个主动传感器应该能发送一个生命刻度，即在特定时间帧中发布状态信息，以便检测出错误动作。
- 每个执行者必须有某种方法调用由**ComputingComponent**计算构件决定的适当应答。
- 每个传感器和执行器应有实施检测其自身操作状态的功能。
- 每个传感器和执行器能测试接受值或发送值的有效性，并且当值超出指定边界时能设定其操作状态。

适用性：对有多个传感器和执行器的任何系统都是非常有用的。

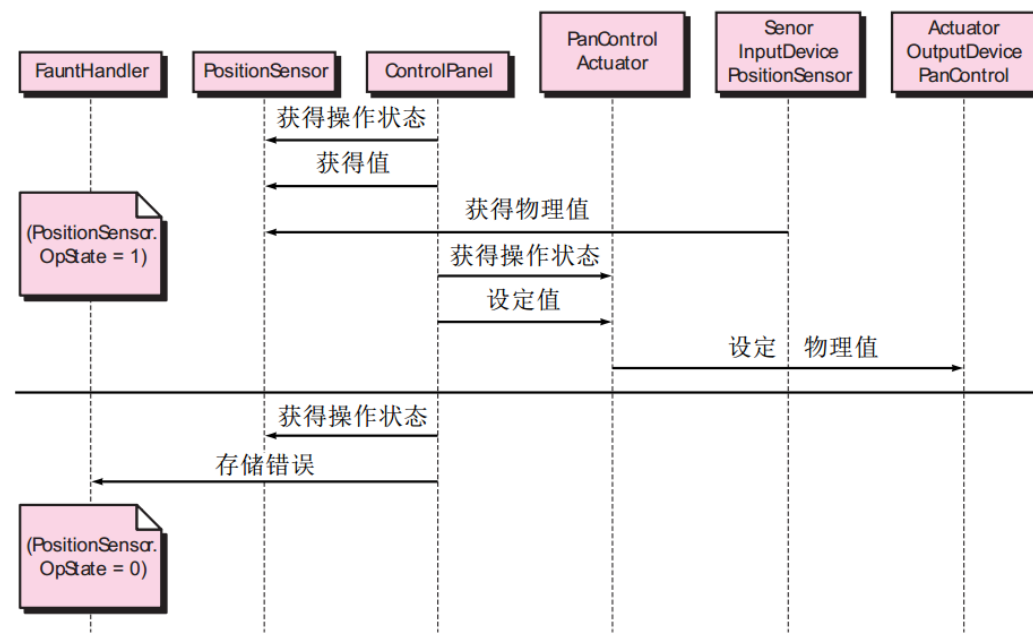
执行器-传感器模式—II

结构体：图10.3显示了**执行器-传感器**模式的UML类图，**执行器**、**被动传感器**和**主动传感器**是抽象类。这个模式中有四种不同的传感器和执行器。传感器和执行器的常见类型为Boolean、Integer和Real。就基本数据类型而言，复杂类是不能用值简单表示的，例如雷达设备。虽然如此但是这些设备还应该继承来自抽象类的接口，因为它们应该具备基本的功能如查询操作状态。



执行器-传感器模式—III

行为：图10-4描述了**执行器-传感器**例子的UML顺序图。这个例子可以应用于*SafeHome*功能，用以控制安全摄像机的调整（例如摇摄、聚焦）。这里在读取或设置一个值时，**ControlPanel**需要一个传感器（被动传感器）和一个执行器（摇摄控制器）进行以诊断为目的的核查操作状态。名为*Set Physical Value*和*Get Physical Value*的消息不是对象间的信息，相反它们描述了系统物理设备和相关软件对应项之间的交互活动。在图的下部即在水平线以下，**PositionSensor**报告操作状态为零。接着**ComputingComponent**发送位置传感器失败的错误代码给**FaultHandler**错误处理程序，它将决定这些错误如何影响系统，并需要采取什么措施。从传感器获得的数据经过计算得到执行器所需的应答。



执行器-传感器模式—III

♥ 其他信息请参看SEPA, 8/e:

- ❖ 参与者
- ❖ 协作
- ❖ 结果



谢谢!

