

本章描述如何来建模一个系统的高层功能使用用例。

## 1.概述

用例描述一个系统的功能，它如何被使用来达到它的多种用户目标。系统的用户被描述通过参与者（actors），其可以表示与系统进行交互的外部系统或人员。

参与者可以使用归纳分类。用例也可使用归纳分类，但是另外一个用例可以包括或扩展其它用例。参与者必须被关联到用例，在这个用例中它们是参与者。正在考虑的系统与参与者之间的关系，和用例通过用例图描述。

用例已经被视为一种机制来捕捉系统需求根据系统的使用。SysML需求可以更明确的捕捉文本需求与用例和其它模型元素之间的关系。在一个用例中，步骤的描述也可以被捕捉作为SysML需求。

不同的方法学以不同的方式使用用例。例如，一些方法需要描述每个用例，对应每个用例绘制在文本中，可以包含前置和后置条件，和主要的、可选的、和/或异常流。用例也可以被进一步详细地阐述描述它们的行为使用活动、交互、或状态机。

## 2.用例图

在用例图中，框架表示一个包或模块，图的内容描述一组参与者和用例之间的关系。对应一个用例图的完整的标题如下：

***uc [model element type] model element name [diagram name]***

用例图的图类型是：***uc***，模型元素类型可以是包或模块。

### 2.1用例图例子

#### 2.1.1使用参与者来表示一个系统的使用

参与者表示使用系统的一个人员、一个组织、或任何外部系统的角色。参与者可以直接或间接通过其它参与者与系统交互。

应该说明的是，参与者是相对的术语，一个参与者可能一个系统的外部元素，但也有可能是另外一个系统的内部元素。例如，假设在组织中的个人请求服务来自组织内部提供IT支持的帮助部门。请求服务的系统和组织的成员被考虑作为帮助部门的参与者。然而，这些相同的个人可能会反过来为一个外部客户提供服务。在这种情形下，以前被认为是相对的帮助台的参与者的个人被认为是“外部”系统的一部分。类似的比喻可以得出一个子系统，内部系统的子系统可以被看作是外部（即一个参与者）到另一个子系统。

参与者使用标准的泛化关系进行分类。参与者分类有一个相似的意义与其它可分类的模型元素的分类。例如，一个特定的参与者参与在所有用例中，更通用的参与者参与其中。

参与者被显示或作为一个贴图下面带有参与者名称，或作为一个矩形包含关键字**«actor»**下面是参与者的名称。标志的选项依赖于使用的工具和使用的方法学。参与者分类被表示使用SysML的标准规范化标志——一条线带有一个中空三角形在范化的终点。

监控系统的用例包包含系统的参与者的描述。五个参与者被显示在图12.2。参与者包含一个**Operator**操作系统和一个**Supervisor**管理系统。**Advanced Operator**角色**Operator**的一个特定的版本，由于角色有附加的特定技巧。注：**Intruder**也被建模作为一个参与者，尽管严格的讲它不是一个用户。入侵者执行与系统的交互是要考虑的外部环境的重要组成部分。同样感兴趣是**Emergency Services**，可能需要报告事件。这个参与者可能已被建模使用一个参与者贴图标志，但因为它不是一个系统和其它设备组成的组织。

### 2.1.2参与者的进一步描述

参与者的属性尽管没有定义在SysML中,多种方法学给出了增加的描述属性的建议，其可以应用到参与者作为一个系统的用户。包含下面的例子:

- 组织，参与者是其一个成员(例，采购)
- 物理位置
- 使用系统所需要的技能水平
- 访问系统所需要的授权证级别

### 2.1.3用例描述系统功能

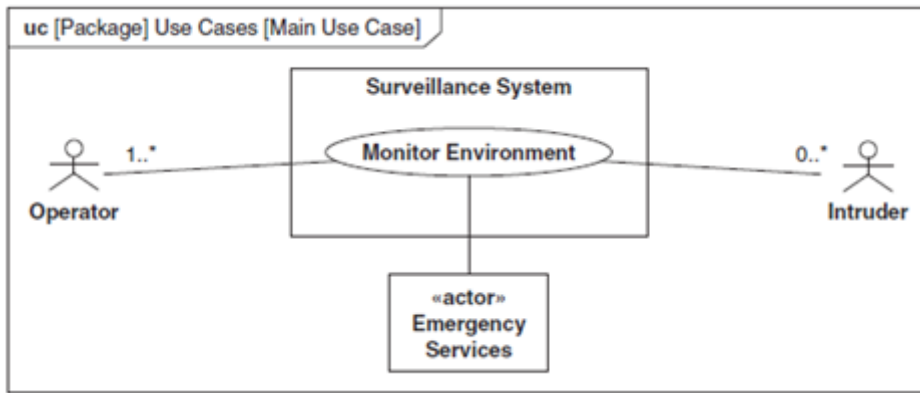
用例从系统使用者的角度描述一个系统的目标。目标被描述根据系统必须支持的功能。典型的,用例描述识别用例的目标,一个主要的使用模式,和许多使用的变体。用例支持的系统提供的功能被称为系统处于考虑中，并常常表示一个系统正在被开发中。系统处于考虑中有时被引用作为主题并被表示通过一个模块。我们将使用术语系统或主题互换的使用来演示处于考虑中的系统。

用例典型的包含许多场景，处于不同的情形中有不同的路径使用用例。

参与者被关联到用例通过通讯路径,被表示作为关联,带有一些限制。关联的终点可以有多重性,在参与者终点的多重性描述每个用例中涉及的参与者的数量。多重性在用例的终点描述用例实例的数目，它的参与者可以被设计在任何给定的时间点。组合关联在每个方向上不被允许；参与者和用例一致被作为成对的。

参与者和用例不可以拥有的属性,所以角色名称在关联上不表示引用属性正如它们可以进行在模块定义图。角色名称在一个参与者终点可以被使用,字面上的描述一个参与者扮演的角色在关联的用例中，无论何时它是不明显的来自参与者的名称。角色名称在用例终点可以被用来描述用例功能如何被关联到相关的参与者。

用例被显示作为一个椭圆形，用例名称在它内部。参与者和用例之间的关联被显示使用标准关联符号。关联终点有默认的多重性,如果没显示,是“0..1”。关联不能有箭头在用例图中，由于没有参与者也没有用例可以拥有属性。一组用例集的主题可以被显示作为一个封闭用例的矩形,带有主题的名称在它的顶部中央。



上图中央显示**Surveillance System**用例,称为**Monitor Environment**。与**Monitor Environment**相关的主要参与者与是系统的**Operator**、**Intruder**和**Emergency Services**。多重性在关联上说明,必须是至少一个**Operator**和潜在的多个**Intruders**。**Emergency Services**也与**Monitor Environment**用例相关,尽管它们不能激活参与者,直到一个**Intruder**被检测和报告。

## 2.2用例之间的关系

用例可以与另一个用例关联使用分类、包含、和扩展关系。

包含 (inclusion) 关系允许一个用例,称为基础用例 (base use case) ,可以包含另外一个用例,另外的一个用例称为包含的用例 (included use case) ,当执行时作为它的功能的部分。当基础用例被执行时, 包含的用例一直被执行。实现基础用例的行为常常引用包含的用例的行为。

明显在包含的定义中, 基础用例的任何参与者可以参与一个包含的用例,所以与一个基础用例相关的一个参与者不需要显式的包含任何用例。例如,正如显示在图12.4, **Operator**隐式参与**Initialize System**和**Shutdown System**通过它们与**Monitor Environment**的相关。

包含用例不试图来表示基础用例的一个功能分解,而是试图来描述通用功能, 可以被包含通过其它用例。在一个功能分解中,底层的功能表示高层功能的一个完整分解。通过对比,一个基础用例和它包含的用例常常描述需要的功能的不同方面。例如, 在图12.4的**Monitor Environment**的用例中, 关键的监控功能被描述通过基础用例,和附加的功能被描述通过包含的**Initialize System**和**Shutdown System**用例。

用例也可以扩展一个基础用例使用**扩展 (extension)** 关系。**扩展用例 (extending use)** 是一个功能的片段, 也即不考虑基础用例功能的部分。它常常描述一些异常行为在交互中, 诸如, 主题和参与者之间的错误处理, 不直接贡献到基础用例的目标。

为了支持扩展, 基础用例定义一组**扩展点 (extension points)** 集, 其表示基础用例可以扩展的位置。扩展点可以被引用作为用例描述的部分。例如,如果有一个序列步骤用例有一个文本描述,扩展点可以被用来说明在那个它的步骤序列中一个扩展的是有效的。一个扩展不得不引用一个扩展点来说明, 在基础用例中, 它可以出现。处于那种扩展的条件是有效的, 可以被进一步描述通过一个约束, 其被评估当扩展点被达到时, 来确定是否扩展的用例发生在这种情形。一个扩展点的存在不隐含, 一个扩展将关联到它。

不像用例包含关系, 基础用例不依赖于扩展用例。然而,扩展用例依赖于它的基础用例中发生的内容; 例如,有可能来假设, 一些异常的情形在基础用例中被发出。没有隐含的意思, 一个参与者与基础用例相

关, 参与在扩展用例中,和扩展用例实际上可以有完全的不同的参与者,正如演示通过**Handle Camera Fault**用例在图12.4。

包含和扩展被显示使用虚线带有一个开放的箭头分别在包含和扩展的终点。包含关系使用关键字«include»和扩展关系使用关键字«extend»。箭头的方向应该被读作**尾部终点包含或扩展箭头终点**。因此,一个包含用例包含一个用例,和一个扩展的用例扩展一个基础用例。一个用例可以有一个附加舱段在它的名称舱段下面,在其中列举它的所有扩展点。扩展线可以有一个附着的注释,其命名他的扩展点并显示条件,在此条件下,扩展用例发生。

分类关系: 用例可以被**分类**使用标准SysML**范化关系**。分类的意义类似于对应其它可分类的模型元素。一个隐含的,例如,场景对应通用的用例也是特定的用例的场景。这也意味着,参与者与一个通用的用例关联可以参与场景描述通过一个特定的用例。用例的分类被显示使用标准SysML范化标志。

图12.4显示用例图包含**Surveillance System**用例的完整的集。作为**Monitor Environment**的部分,常规的**Operators**仅被允许来监控可疑的运动的自动追踪—也即是,当系统控制摄像头时。这允许安全公司使用初级或比较少培训的工程师对应这个目标。**Advanced Operators**可以参与进**Manually Monitor Environment**用例,当它们控制摄像头正如手工的使用一个游戏手柄。**Advanced Operators**也有选项来设置监控摄像头的轨迹作为下面的。注: 尽管根据用例分类规则**Operators**可以参与**Manually Monitor Environment**,在这种情况下,正如通常发生的,它的主场景指定操作, 仅一个特定的参与者(**Advanced Operator**)可以进行。

**Monitor Environment**的完整需求规格也包含系统初始化和关闭正如指示提供**Monitor Environment**和**Initialize System**和**Shutdown System**的包含关系。

**Fault**扩展点表示一个位置在**Monitor Environment**用例中, 其中摄像头失效可以被处理。**Handle Camera Fault**用例扩展**Monitor Environment**在**Fault**扩展点。它是一个异常任务, 其将被触发当摄像头失效被检测到,正如说明通过它的关联条件,和可以仅被执行通过**Supervisor**。

基于文本的**用例描述 (use case description)** 可以被用来提供附加信息来支持用例定义。这个描述明显的贡献到用例的值。描述文本可以被绘制在模型中作为一个单一或多个注释。也有可能对待每个步骤在一个用例描述作为一个SysML需求。一个典型的用例的描述可以包含下面的内容:

- 前处理条件: 必须持有的条件对应用例开始。
- 后处理条件: 一旦用例已经完成, 必须持有的条件。
- 主要流: 最常见的场景或用例的场景。
- 备选和/或异常流: 很少见的场景或非正式关闭。异常流可以引用扩展点并通常表示流, 不直接处于主要流的支持目标中。

其它信息可以增强基础用例描述来进一步详细描述参与者和主题之间的交互。

这里是一个抽取来自监控环境的用例描述:

- 前处理条件

**Surveillance System是关闭的。**

- 主要流 (Primary Flow)

**Operator 或 Operators将使用Surveillance System来监控设施处于监控下的环境。一个Operator将初始化系统(参考Initialize System), 在操作和关闭系统之前(参考Shutdown System)。在常规的操作**

中,系统的摄像头将自动追溯预先设置的路径, 已经被设置为优化监测的可能性。

如果一个Intruder被检测到,一个警报将被发起在内部或使用一个中央监控站,它的职责它是来召唤任何所需的辅助。如果有一个聪明的入侵者追踪系统,其将重载标准摄像头搜索路径,将被鼓励在这个点追踪可疑的入侵者。如果没有随后期望操作符将保持可疑的入侵者可视的轨迹并传递这个这个知识到Emergency Services, 如果和何时它们到达。

- 备选的流程 (Alternate Flow)

在系统初始化之后,但在正式操作开始前,可能有一个即刻的一个失效将发起,在这种情形下它可以被处理, (参考Fault的扩展点),但失效将不处理从那以后

- 后处理条件

**Surveillance System被关闭。**

用例与行为的详细说明

用例的文本定义,与先前描述的用例模型一起,可以描述一个系统的功能。然而,如果期望,用例的更详细定义可以被建模使用交互、活动、或状态机,描述在第9章到第11章。典型的这些定义被添加,在用例定义已经被检查和定义已经被审查之后并达成详细说明的需求和设计。行为的形式主义的选项常常是个人的或项目选择的,但通常:

- 当一个场景主要是基于消息机制时,交互是非常有用的。
- 场景包含考虑数据传输的控制逻辑、输入和输出流和/或算法时,活动是非常有用的。
- 状态机是有用的,当交互在参与者和主题之间是异步的并不能简单的表示通过一个顺序化的时间序列。

语境图

使用交互或活动时,生命线和分区表示用例中的参与者。生成一个内部模块图是非常有用的,其中封装框架对应**系统语境 (system context)**,主题和参与者对应内部模块图中的组成部分。为了支持这种技术,参与者可以出现在一个模块定义图中,和一个组成部分在一个内部模块图可以键入通过参与者。可选的是,参与者可以被分配到模块使用分配关系描述在第14章,和随后表示参与者的组成部分可以键入通过模块。

图12.5显示内部模块图,其描述模块**System Context**的内部结构,其表示**Surveillance System**的语境和它相关的用例。系统处于考虑中, **Surveillance System**被表示作为**System Context**的组成部分,称为**company security system**。两个参与者 **Advanced Operator**和**Intruder**, 它们参与在用例中,也分别表示作为**security guard**和**suspected thief**组成部分。

序列图

一个用例,除了被描述在一个用例描述中,可以被详细说明描述通过序列图中的一个或多个交互。不同的交互可以对应于基础用例,任何包含的用例,和许多扩展用例。拥有交互的模块必须有组成部分,对应主题和参与者,可以随后被表示通过序列图中的生命线。

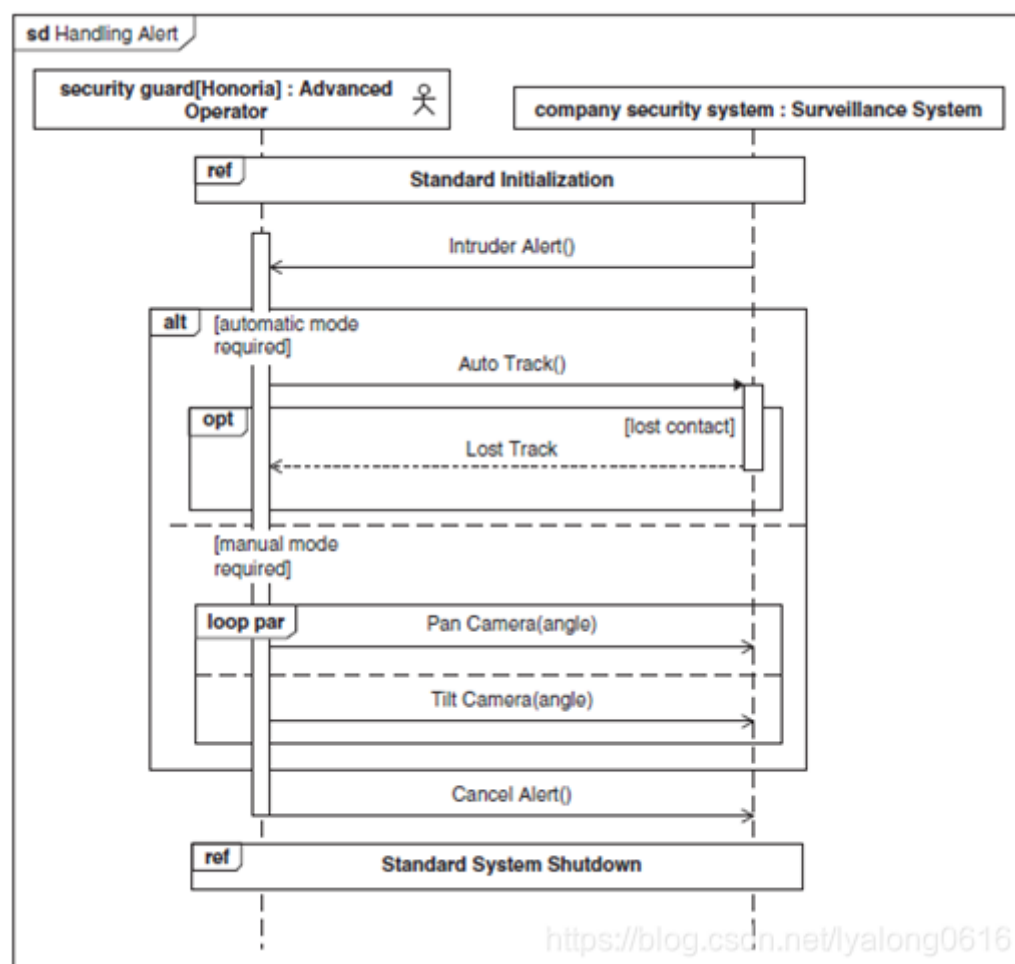
正如前面陈述的,包含用例必须一直发生作为它的基础用例的部分。作为一个结果,一个交互描述一个包含的场景将典型的是一个基础场景表示的一个交互的强制性部分。典型的被说明在基础交互场景内部,通过引用交互对应包含的场景在组合片段的内部带有一个操作符,诸如, **seq,strict**,或**loop**。

严格的讲,一个交互表示一个基础用例应该被指定不需要引用到扩展的用例,简单的注意扩展点。然而,一个大众化的方法是来引用扩展的用例作为可选的构件在交互中,表示基础应用场景。在这种方法中,一

个交互对应一个扩展的用例典型的包含在一个条件操作符的一个操作数,诸如, **break**、**opt**或**alt**。操作数应该被守卫使用扩展上的约束,如果一个被指定。

模块**System Context**的内部模块图被显示在图12.5,拥有许多交互。交互描述**Manually Monitor Environment**用例的主要场景, **Handling Alert**被显示在图12.6。在图12.4中, **Manually Monitor Environment**用案包含**Initialize System**用例和**Shutdown System**用例。 **Handling Alert**交互包含相应的交互**Standard Initialization**的使用, 其是一个场景对应**Initialize System**用例,和交互**Standard Shutdown**, 其是一个场景对应**Shutdown System**用例。

在这两种交互之间, 场景描述看门狗**Honor***a*,如何处理一个入侵者警报。由于她是一个**Advanced Operator**,她可以手工的控制摄像头正如它希望的,或她可以选择允许系统来自动跟踪怀疑的入侵者。交互对应更通用的用例**Monitor Environment**显示在图12.4,不会包含人工控制摄像头。



活动图

正如先前提及的,一个用例场景也可被表示通过一个活动图,在这种情形中参与者被表示作为活动分区。使用交互,一个活动可以详细说明一个基础用例、包含用例、和扩展用例。

图12.7显示一个可选的描述, 如何人工跟踪怀疑的入侵者被处理对应**Manually Monitor Environment**用例。两个活动分区表示**security guard**和**company security system**,被使用来说明那个用例参与者具有责任对应那个动作。

新的入侵者情报(来自什么源, 我们没有告知)被分析。控制流初始化通过情报的接收, 被分支来解决两个关注。如果入侵者移动位置,随后一个**Move Joystick**动作被执行来跟踪入侵者。如果入侵者出现已经移动出当前摄像头的范围, 随后一个**Select Camera**动作被执行来选择一个更合适的摄像头。在两

种情形中, 当没有动作被需要时一个流的最终节点被使用来处理情形。同时,这个流的输入被转换成**Pan Camera**和**Tilt Camera**消息到恰当的摄像头通过**Issue Camera Commands**动作。

### 状态机图

状态机也可以被使用来表示场景,尽管一些方法鼓励使用一个单一的状态机来表示用例的所有可能场景,包含异常情况。注:当使用一个状态机时,没有构件,诸如,交互生命线或活动划分,来明确标识组成部分的职责采取动作。然而,状态机,交互通过信号交互可以被定义对应每个分区,包含感兴趣的系统和参与者。

图12.8显一个状态机的部分描述**Manually Monitor Environment**用例。它显示了三个状态**operator idle**、**intruder present**和**automatic tracking enabled**。当处于**operator idle**时,一个**Intruder Alert**事件引起**Raise Alarm**消息将被发送,和一个转变到**intruder present**状态。一旦处于**intruder present**状态,入侵者可以被人工追踪,到一个**Auto Track**事件将触发一个转变到**automatic tracking enabled**并禁止手工追踪直到一个**Lost Track**事件发生。

这个描述分享许多信号与图12.6,但它聚焦在状态上而不是消息。

## 小结

用例被使用来捕捉一个系统需要达到用户目标的功能。一个用例被常常用作一种方式来描述一个系统的功能需要并可以增强SysML 需求来进一步细化基于文本的功能需求的定义。用例被使用的方式是严重依赖方法学的。下面是关键的用例概念介绍在本章。

- 用例描述一个系统的一个特定使用来获得一个期望的用户目标。用例关系对应包含、扩展、和分类是有用的,将常见的功能分解到用例中,可以被重用通过其它用例。一个包含的用例一直被执行作为基础用例的部分。一个用例扩展基础用例通常被执行通过异常,并通常没有在基础用例的直接支持目标中。
- 系统处于考虑中(也被称为主题)提供参与者需要的功能,表示作为用例。
- 参与者描述一个角色扮演通过一个系统的外部实体可以表示人员、组织、或顶层外部系统。总的可以被使用来表示不同分类的参与者之间的分类关系。关联参与者到它们参与其中的用例。
- 功能描述通过一个用例常常被使用更详细的交互、活动、和状态机说明。行为形式的选择和它们如何使用常常依赖于特定的方法学。