

## 需求建模：基于场景的方法

## 要点浏览

**概念：**文字记录是极好的交流工具，但并不一定是表达计算机软件需求的最好方式。需求建模使用文字和图表的综形式，以相对容易理解的方式描绘需求，更重要的是，可以更直接地评审它们的正确性、完整性和一致性。

**人员：**软件工程师（有时被称作分析师）使用从客户那里获取的需求来构建模型。

**重要性：**为了确认软件需求，你需要从不同的视角检验需求。本章将从基于场景的观点考虑需求建模并检验如何在UML中使用补充场景。在第10章和第11章会学习需求建模的其他“维度”。在检验大量不同维度时，会增加发现错误的概率。这些错误可能是与表象不一致，或可能是没有发现的缺失项。

**步骤：**基于场景的建模从用户的角度表现系统。在基于场景建模时，将更好地理解用户如何与软件交互，发现没有覆盖到的利益相关者所需的主要系统功能和特性。

**工作产品：**基于场景建模产生的面向文本的表达称作“用例”。用例描述了特定交互方式，形成非正式（只简单描述）或更加结构化或正规化的自然特征。这些用例能补充大量不同的UML图，覆盖更多交互的程序化观点。

**质量保证措施：**必须评审需求建模工作产品的正确性、完整性和一致性，必须反映所有利益相关者的要求并为从中导出设计建立基础。

在技术层面上，软件工程开始于一系列的建模工作，最终生成待开发软件的需求规格说明和设计表示。需求模型实际上是一组模型<sup>①</sup>，是系统的第一个技术表示。

在一本关于需求建模方法的开创性书籍中，Tom DeMarco[Dem79]如下这样描述该过程：

回顾分析阶段的问题和过失，我建议对分析阶段的目标进行以下的增补。分析的结果必须是高度可维护的，尤其是要将此结果应用于目标文档（软件需求规格说明）。必须使用一种有效的分割方法解决规模问题，维多利亚时代小说式的规格说明是不行的。尽可能使用图形符号。考虑问题时必须区分逻辑的（本质）和物理的（实现）……无论如何，我们至少需要……某种帮助我们划分需求的方法，并在规格说明前用文档记录该划分……某种跟踪和评

## 关键概念

活动图  
域分析  
正式用例  
需求分析  
需求建模  
基于场景建模  
泳道图  
UML 模型  
用例  
用例异常

166

① 本书过去的版本使用分析模型这个术语而不是需求模型。本版中决定使用这两个术语，以便表达在解决问题的不同方面时定义的建模活动。分析是获取需求时的动作。

估接口的手段……使用比叙述性文本更好的新工具来描述逻辑和策略……

尽管 DeMarco 在 25 年前就写下了关于分析建模的特点,但他的意见仍然适用于现代的需求建模方法和表示方法。

## 9.1 需求分析

需求分析产生软件工作特征的规格说明,指明软件和其他系统元素的接口,规定软件必须满足的约束。在需求分析过程中,软件工程师(有时这个角色也被称作分析师或建模师)可以细化在前期需求工程的起始、获取、协商任务中建立的基础需求(第 8 章)。

需求建模动作结果为以下一种或多种模型类型:

- 场景模型:出自各种系统“参与者”观点的需求。
- 面向类的模型:表示面向对象类(属性和操作)的模型,其方式为通过类的协作获得系统需求。
- 基于行为和模式的模型:描述如何将软件行为看作外部“事件”后续的模式。
- 数据模型:描述问题信息域的模型。
- 面向流的模型:表示系统的功能元素并且描述当功能元素在系统中运行时怎样进行数据变换。

这些模型为软件设计者提供信息,这些信息可以被转化为结构、接口和构件级的设计。最终,在软件开发完成后,需求模型(和需求规格说明)就为开发人员和客户提供了评估软件质量的手段。

167

本章关注基于场景的建模,这项技术在整个软件工程界发展迅猛。在第 10 章和第 11 章,我们考虑基于类的模型和行为模型。在过去十几年,人们已经不常使用流和数据建模,而逐步流行使用场景和基于类的方法,以此作为行为方法和基于模式技术的补充。<sup>①</sup>

### 9.1.1 总体目标和原理

在整个分析建模过程中,软件工程师的主要关注点集中在做什么而不是怎么做。在特定环境下发生哪些用户交互?系统处理什么对象?系统必须执行什么功能?系统展示什么行为?定义什么接口?有什么约束?<sup>②</sup>

在前面的章节中,我们注意到在该阶段要得到完整的需求规格说明是不可能的。客户也许无法精确地确定想要什么,开发人员也许无法确定能恰当地实现功能和性能的特定方法,这些现实情况都削弱了迭代需求分析和建模方法的效果。分析师将为已经知道的内容建模,并使用该模型作为软件进一步扩展的设计基础。<sup>③</sup>

需求模型必须实现三个主要目标:(1)描述客户需要什么;(2)为软件设计奠定基础;

**引述** 任何一个需求“视图”都不足以理解和描述一个复杂系统所需的行为。

Alan M. Davis

**关键点** 一旦软件完成后,分析模型和需求规格说明书将成为评估软件质量的手段。

**引述** 需求不是架构。需求既不是设计,也不是用户接口。需求就是指需要什么。

Andrew Hunt,  
David Thomas

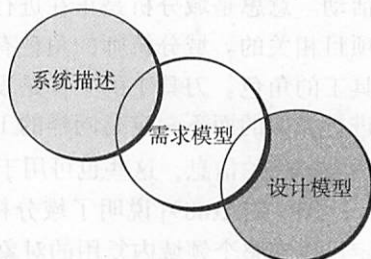
① 在这一版本中我们省略了面向流建模和数据建模。但是在网上可以找到这些较老的需求建模方法的大量信息。如果你感兴趣,可以搜索关键词“结构化分析”进行查找。

② 应该注意,当客户变得更加精通技术时,规格说明书中的“怎么做”需同“做什么”一样重要。但是,基本关注点应保留在“做什么”上。

③ 软件团队也可以花些功夫选择生成一个原型(第 4 章),以便更好地理解系统的需求。

(3) 定义在软件完成后可以被确认的一组需求。分析模型在系统级描述和软件设计(第12~18章)之间建立了桥梁。这里的系统级描述给出了在软件、硬件、数据、人员和其他系统元素共同作用下的整个系统或商业功能,而软件设计给出了软件的应用程序结构、用户接口以及构件级的结构。这个关系如图9-1所示。

重要的是要注意需求模型的所有元素都可以直接跟踪到设计模型。通常难以清楚地区分这两个重要的建模活动之间的分析和设计工作,有些设计总是作为分析的一部分进行,而有些分析将在设计中进行。



168

### 9.1.2 分析的经验原则

Arlow 和 Neustadt[Arl02] 提出了大量有价值的经验原则,在创建分析模型时应该遵循这些经验原则:

- 模型应关注在问题域或业务域内可见的需求,抽象的级别应该相对高一些。“不要陷入细节”[Arl02],即不要试图解释系统如何将工作。
- 需求模型的每个元素都应能增加对软件需求的整体理解,并提供对信息域、功能和系统行为的深入理解。
- 关于基础结构和其他非功能的模型应推延到设计阶段再考虑。例如,可能需要一个数据库,但是只有在已经完成问题域分析之后才应考虑实现数据库所必需的类、访问数据库所需的功能以及使用时所表现出的行为。
- 最小化整个系统内的关联。表现类和功能之间的联系非常重要,但是,如果“互联”的层次非常高,则应该想办法减少互联。
- 确认需求模型为所有利益相关者都带来价值。对模型来说,每个客户都有自己的使用目的。例如,业务人员将使用模型确认需求,设计人员将使用模型作为设计的基础,质量保证人员将使用模型帮助规划验收测试。
- 尽可能保持模型简洁。如果没有提供新的信息,就不要添加附加图表;如果一个简单列表够用,就不要使用复杂的表示方法。

图9-1 需求模型在系统描述和设计模型之间建立桥梁

**关键点** 分析模型应该描述什么是客户所需,应该建立设计的基础,建立有效的目标。

**提问** 进行需求分析时有没有可以帮助我们指南?

**引述** 值得进攻的问题总是通过反击证明其价值。  
Piet Hein

169

### 9.1.3 域分析

在需求工程讨论中(第8章),我们注意到分析模式通常在特定业务领域内的很多应用系统中重复发生。如果用一种方式对这些模式加以定义和分类,让软件工程师或分析师识别并复用这些模式,将促进分析模型的创建。更重要的是,应用可复用的设计模式和可执行的软件构件的可能性将显著增加。这将把产品投放市场的时间提前,并减少开发费用。

但问题是,首先如何识别分析模式?由谁来对分析模式进行定义和分类,并为随后的项目准备好分析模式?这些问题的答案在域分析中。Firesmith[Fir93]这样描述域分析:

软件域分析是指识别、分析和详细说明某个特定应用领域的共同需求,特别是那些在该应用领域内被多个项目重复使用的……(面向对象的域分析是)在某个特定应用领域内,根

**网络资源** 很多关于域分析的有用信息可以在 [www.sei.cmu.edu](http://www.sei.cmu.edu) 中找到。

据通用的对象、类、部件和框架，识别、分析和详细说明公共的、可复用的能力。

“特定应用领域”的范围从航空电子设备到银行业，从多媒体视频游戏到医疗设备中的嵌入式软件。域分析的目标很简单：查找或创建那些广泛应用的分析类或分析模式，使其能够复用。<sup>①</sup>

使用本书前面介绍的术语，域分析可以被看作软件过程的一个普适性活动。意思是域分析是正在进行的软件工程活动，而不是与任何一个软件项目相关的。域分析师的角色有些类似于重型机械制造业中一名优秀的刀具工的角色。刀具工的工作是设计并制造工具，这些工具可被很多人用来进行类似的而不一定是同样的工作。域分析师<sup>②</sup>的角色是发现和定义可复用的分析模式、分析类和相关信息，这些也可用于类似但不要求必须是完全相同的应用。

图 9-2[Arn89] 说明了域分析过程的关键输入和输出。应该调查领域知识的来源以便确定可以在整个领域内复用的对象。

**关键点** 域分析不关注特定的应用系统，而是关注应用所属的领域。其目的在于识别那些解决可用于域内所有应用系统的共同问题。

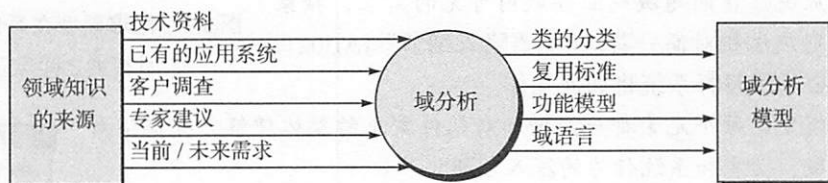


图 9-2 域分析的输入和输出

## SafeHome 域分析

**[场景]** Doug Miller 的办公室，在销售业务会议之后。

**[人物]** Doug Miller，软件工程经理；Vinod Raman，软件工程团队成员。

**[对话]**

**Doug**：我需要你做一个特殊的项目，Vinod。我将会把你调离需求收集会议。

**Vinod**（皱眉不悦）：太糟了。这可行吗……我已从中获取了一些需求。出什么事啦？

**Doug**：Jamie 和 Ed 将接管你的工作。不管怎样，市场部坚持要我们在第一次发布的 SafeHome 中交付具有互联网能力的家庭安全功能。我们一直紧张地为此工作着……

没有足够的时间和人力，所以我们马上就要解决 PC 接口和 Web 接口两个问题。

**Vinod**（看上去很疑惑）：我不知道原先设定的计划……我们甚至还没有完成需求收集。

**Doug**（无精打采地微笑）：我知道，但时间太紧了，我决定马上和市场部开始战略合作……无论如何，一旦从所有需求收集会议上获得信息，我们就将重新审视任何不确定的计划。

**Vinod**：好的，会发生什么事？你要我做什么事吗？

**Doug**：你知道“域分析”吗？

**Vinod**：略知一些。在建立应用系统时为

① 域分析的一个补充观点是：“包括为域建模，因此软件工程师和其他的利益相关者可以更好地学习……不是所有域的都必然导致可复用的类。” [Let03]。

② 不要认为有域分析员在工作，软件工程师就不需要理解应用问题的领域。软件团队的每个成员都应该一定程度地了解软件将要工作的领域。



做同一件事的应用系统寻找相似的模式。如果可能，可以在工作中剽窃这些模式并复用它们。

**Doug**：我觉得用剽窃这个词不太恰当，但你的意思基本是正确的。我想让你做的事是开始研究可控制 SafeHome 这类系统的现存用户接口。我想你需要组织一套模式和分析类，它们通常既能坐在房间里处理基于 PC 的接口，也能处理通过互联网进入的基于浏览器的接口。

**Vinod**：把它们做成相同的东西可以节省时间……为什么不这么做呢？

**Doug**：哦……. 有你这种想法的人真是非

常好。整个核心要点是如果能识别出两种接口，那么就采用相同的代码，等等，这样我们就节省了时间和人力。这些正是市场部坚持的。

**Vinod**：那你想要什么？类，分析模式，设计模式？

**Doug**：所有。非正式地说这就是关键所在。我就是想稍早一些开始我们的内部分析和设计工作。

**Vinod**：我将在类库中看看我们已经得到了哪些。我也会使用几个月前我读的一本书中的模式模版。

**Doug**：太好了，继续工作吧。

#### 9.1.4 需求建模的方法

一种考虑数据和处理的需求建模方法称作结构化分析，其中处理过程将数据作为独立实体加以转换。数据对象建模定义了对对象的属性和关系，操作数据对象的处理建模应表明当数据对象在系统内流动时处理过程将如何转换数据。

需求建模的第二种方法称作面向对象的分析，这种方法关注类的定义和影响客户需求的类之间的协作方式。UML 和统一过程（第 4 章）主要是面向对象的分析方法。

在本书这一版中，我们已经选择强调采用面向对象分析的元素进行 UML 建模。目的是建议一种联合的表达方式，给项目利益相关者提供最好的软件需求，以便为软件设计提供桥梁。

如图 9-3 所示，需求模型的每个元素表示源自不同观点的问题。基于场景的元素表述用户如何与系统和使用软件时出现的特定活动序列进行交互。基于类的元素的内容包括：系统操作的对象，应用在这些对象间影响操作和对象间关系（某层级）的操作，以及定义类间发生的协作。行为元素描述了外部事件如何改变系统或驻留在系统里的类的状态。最后，面向流的元素表示信息转换的系统，描述了数据对象在流过各种系统功能时是如何转换的。

需求模型导出每个建模元素的派生类。然而，每个元素（即用于构建元素和模型的图表）的特定内容可能因项目而异。就像我们在本书中多次提到的那样，软件团队必须想办法保持模型的简单性。只有那些为模型增加价值的建模元素才能使用。

**引述** 分析容易使人灰心丧气，全都是非常复杂的人际关系，不确定且困难的东西。总而言之，分析让人着迷。一旦沉迷，原来轻松构建系统的快乐将难以令你满足。

Tom DeMarco

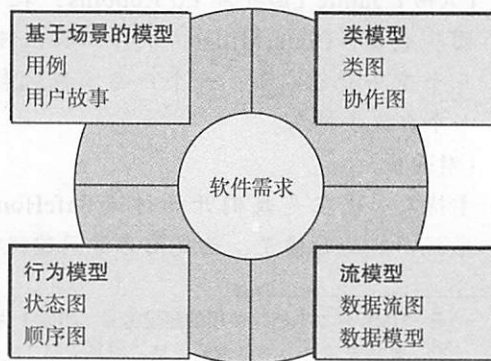


图 9-3 需求模型的元素

## 9.2 基于场景建模

尽管可以用多种方式度量基于计算机的系统或产品的成果，但用户的满意度仍是其中最重要的。如果软件工程师了解最终用户（和其他参与者）希望如何与系统交互，软件团队将能够更好、更准确地刻画需求特征，完成更有针对性的分析和设计模型。因此，使用 UML<sup>①</sup>需求建模将从开发用例、活动图和泳道图形式的场景开始。

**提问** 在描述需求模型时常用哪些不同的观点？

### 9.2.1 创建初始用例

Alistair Cockburn 刻画了一个名为“合同行为”的用例 [Coc01b]。我们在第 8 章讨论的“合同”定义了一个参与者<sup>②</sup>使用基于计算机的系统完成某个目标的方法。本质上用例捕获了信息的产生者、使用者和系统本身之间发生的交互。在本节，我们研究如何开发用例，这是分析建模活动的一部分<sup>③</sup>。

**引述**（用例）只是帮助定义系统之外（参与者）存在什么以及系统应完成什么（用例）。

Ivar Jacobson

第 8 章中我们已经提到，用例从某个特定参与者的角度出发，采用简明的语言描述一个特定的使用场景。但是我们如何知道：（1）编写什么？（2）写多少？（3）编写说明应该多详细？（4）如何组织说明？如果想让用例像一个需求建模工具那样提供价值，那么必须回答这些问题。

**关键点** 在某些情况下，用例成为最主要的需求工程机制，但是这并不意味着你应该放弃适用的其他建模方法。

**编写什么？** 两个首要的需求工程工作——起始和获取——提供了开始编写用例所需要的信息。运用需求收集会议、质量功能部署（Quality Function Deployment, QFD）和其他需求工程机制确定利益相关者，定义问题的范围，说明整体的运行目标，建立优先级顺序，概述所有已知的功能需求，描述系统将处理的信息（对象）。

开始开发用例时，应列出特定参与者执行的功能或活动。这些可以借助所需系统功能的列表，通过与利益相关者交流，或通过评估活动图（作为需求建模中的一部分而开发）获得（9.3.1 节）。

173

### SafeHome 开发另一个初始用户场景

**[场景]** 会议室，第二次需求收集会议中。

**[人物]** Jamie Lazar 和 Ed Robbins，软件团队成员；Doug Miller，软件工程经理；三个市场营销人员；一个产品工程代表；一个会议主持人。

**[对话]**

**主持人：**现在我们开始讨论 SafeHome 监视功能的时候了，让我们为访问监视功

能开发一个用户场景。

**Jamie：**谁在其中扮演参与者的角色？

**主持人：**我想 Meredith（市场营销人员）已经在该功能上进行了一些工作，你来试试这个角色吧。

**Meredith：**你想采用我们上次用的方法，是吗？

**主持人：**是的，同样的方法。

① UML 是本书通篇使用的建模符号。附录 1 为那些不熟悉 UML 基本符号的读者提供了简要指南。

② 参与者不是一个确定的人员，而是人员（或设备）在特定的环境内所扮演的一个角色，参与者“呼叫系统并由系统提供一种服务” [Coc01b]。

③ 用例是用户接口的分析建模中特别重要的一部分，我们将在第 15 章详细讨论接口分析。

**Meredith**：好的，很明显，开发监视功能的理由是允许房主远距离检查房屋、记录并回放捕获的录像……就是这样。

**Ed**：我们采用压缩的方法存储图像吗？

**主持人**：好问题，但是我们现在先不考虑实现的问题，Meredith，你说呢？

**Meredith**：好的，这样对于监视功能基本上就有两部分……第一部分是配置系统，包括布置建筑平面图——我们需要工具来帮助房主做这件事，第二部分是实际的监视功能本身。因为布局是配置活动的一部分，所以我将重点集中在监视功能。

**主持人（微笑）**：抢先说出我想说的话。

**Meredith**：哦……我希望通过电脑或通过 Internet 访问监视功能。我的感觉是 Internet 访问可能使用的频率更高一些。

不管怎样，我希望能够在计算机上和控制面板上显示摄像机图像并移动某个摄像机镜头。在房屋平面设计图上可以选择指定摄像机，我希望可以有选择地记录摄像机输出和回放摄像机输出，我还希望能够使用特殊的密码阻止对某个或多个摄像机的访问。希望有支持小窗口显示形式的选项，即从所有的摄像机显示图像，并能够选择某一个进行放大。

**Jamie**：那些叫作缩略视图。

**Meredith**：对，然后我希望从所有摄像机获得缩略视图。我也希望监视功能的接口和所有其他的 SafeHome 接口有相同的外观和感觉。

**主持人**：干得好，现在，让我们更详细地讨论这个功能……

上面讨论的 SafeHome 住宅监视功能（子系统）确定了如下由参与者房主执行的功能（简化列表）：

- 选择将要查看的摄像机。
- 提供所有摄像机的缩略视图。
- 在计算机的窗口中显示摄像机视图。
- 控制某个特定摄像机的镜头转动和缩放。
- 可选择地记录摄像机的输出。
- 回放摄像机的输出。
- 通过 Internet 访问摄像机监视功能。

174

随着和利益相关者（扮演房主的人）交谈的增多，需求收集团队将为每个标记的功能开发用例。通常，用例首先用非正式的描述性风格编写。如果需要更正式一些，可以使用类似于第8章中提出的某个结构化的形式重新编写同样的用例，在本节的后面我们将进行重新生成。

为了举例说明，考虑“通过互联网访问摄像机监视设备—显示摄像机视图（Access Camera Surveillance-Display Camera Views, ACS-DCV）”功能，扮演参与者房主的利益相关者可能会编写如下说明。

**用例：通过互联网访问摄像机监视设备—显示摄像机视图（ACS-DCV）。**

**参与者：房主。**

如果我正在进行远程访问，那么我可以使用任何计算机上的合适的浏览器软件登录 SafeHome 产品网站。输入我的账号和两级密码，一旦被确认，我可以访问已安装的 SafeHome 系统的所有功能。为取得某个摄像机视图，从显示的主功能按钮中选择“监视”，然后选择“选取摄像机”，这时将会显示房屋的平面设计图，之后再选择感兴趣的摄像机。另一种可选方法是，通过选择“所有摄像机”同时从所有的摄像机查看缩略视图快照。当选

择了某个摄像机时，可以选择“查看”，然后以每秒一帧速度显示的图像就可以在窗口中显示。如果希望切换摄像机，则选择“选取摄像机”，这时原来窗口显示的信息消失，并且再次显示房间的平面设计图，然后就可以选择感兴趣的摄像机，以便显示新的查看窗口。

描述性用例的一种表达形式是通过用户活动的顺序序列表现交互，每个行动由声明性的语句表示。再以 ACS-DCV 功能为例，我们可以写成如下形式。

用例：通过互联网访问摄像机监视设备 - 显示摄像机视图 (ACS-DCV)。

参与者：房主。

1. 房主登录 SafeHome 产品网站。
2. 房主输入账号。
3. 房主输入两个密码（每个至少 8 个字符长度）。
4. 系统显示所有的主要功能按钮。
5. 房主从主要功能按钮中选择“监视”。
6. 房主选择“选取摄像机”。
7. 系统显示房屋的平面设计图。
8. 房主从平面设计图中选择某个摄像机图标。
9. 房主选择“视图”按钮。
10. 系统显示一个由摄像机编号确定的视图窗口。
11. 系统在视图窗口内以每秒一帧的速度显示视频输出。

注意，这个连续步骤的陈述没有考虑其他可能的交互（描述更加自由随意而且确实表达了一些其他选择）。这种类型的用例有时被称作主场景 [Sch98a]。

## 9.2.2 细化初始用例

为了全面理解用例描述功能，对交互操作给出另外的描述是非常有必要的。

因此，主场景中的每个步骤将通过如下提问得到评估 [Sch98a]：

- 在这一步，参与者能做一些其他动作吗？
- 在这一步，参与者有没有可能遇到一些错误条件？如果有可能，这些错误会是什么？
- 在这一步，参与者有没有可能遇到一些其他行为（如由一些参与者控制之外的事件调用）？如果有，这些行为是什么？

这些问题的答案导致创建一组次场景，次场景属于原始用例的一部分，但是表现了可供选择的行为。例如，考虑前面描述的主场景的第 6 和第 7 步。

6. 房主选择“选取摄像机”。
7. 系统显示房屋的平面设计图。

在这一步，参与者能做一些其他动作吗？答案是肯定的。考虑自由随意的操作方式，参与者可以选择同时查看所有摄像机的缩略视图。因此，一个次场景可能是“查看所有摄像机的缩略视图”。

在这一步，参与者有没有可能遇到一些错误条件？作为基于计算机的系统操作，任何数量的错误条件都可能发生。在该语境内，我们仅仅考虑在第 6 和第 7 步中说明的活动的直接错误条件，问题的答案还是肯定的。带有摄像机图标的房屋平面图可能还没有配置过，这样

**引述：**用例可以应用在许多（软件）过程中，我们感兴趣的是迭代和风险驱动过程。

Geri Schneider,  
Jason Winters

**提问：**在开发场景用例时，如何检查动作的可选过程？



选择“选取摄像机”就导致错误的条件：“没有为该房屋配置平面设计图”<sup>①</sup>。该错误条件就成为一个次场景。

176

在这一步，参与者有没有可能遇到一些其他行为？问题的答案再一次是肯定的。当第6和第7步发生时，系统可能遇到报警。这将导致系统显示一个特殊的报警通知（类型、地点、系统动作），并向参与者提供和报警性质相关的一组操作。因为这个次场景可以在所有的实际交互中发生，所以不会成为 ACS-DCV 用例的一部分。而且，我们将开发一个单独的用例——遇到报警条件——这个用例可以被其他用例引用。

前面段落描述的每种情景都是以客户用例的异常处理为特征的。异常处理描述了这样一种情景（可能是失败条件或参与者选择了替代方案），该场景导致系统展示出某些不同的行为。

**提问** 什么是用例异常？我们如何决定这些异常的样子？

Cockburn[Coc01b] 推荐使用“头脑风暴”来推动团队合理地完成每个用例中一系列的异常处理。除了本节前面提到了三个常规问题外，还应该研究下面的问题：

- 在这个用例中是否有某些具有“确认功能”的用例出现？包括引用确认功能，以及可能出现的出错条件。
- 在这些用例中是否有支持功能（或参与者）的应答失败？例如，某个用户动作是等待应答，但该功能已经应答超时了。
- 性能差的系统是否会导致无法预期或不正确的用户活动？例如，一个基于 Web 的接口应答太慢，导致用户在处理按钮上已经做了多重选择。这些选择队列最终不恰当地生成了一个出错条件。

其他问题和回答将继续扩充这份列表，使用下面的标准可使这些问题合理化 [Co01b]：用例应该注明异常处理，即如果软件能检测出异常所发生的条件就应该马上处理这个条件。在某些情况下，异常处理可能拖累其他用例处理条件的开发。

### 9.2.3 编写正式用例

9.2.1 节表述的非正式用例对于需求建模常常是够用的。但是，当用例需要包括关键活动或描述一套具有大量异常处理的复杂步骤时，我们就会希望采用更为正式的方法。

在 SafeHome 中的 ACS-DCV 用例把握了正式用例的典型描述要点。在以下的 SafeHome 中：情境目标确定了用例的全部范围。前提条件描述在用例初始化前应该知道哪些信息。触发器确定“用例开始”的事件或条件 [Coc01b]。场景列出参与者和恰当的系统应答所需要的特定活动。异常处理用于细化初始用例时没有涉及的情景（9.2.2 节）。此外还可能包含其他主题，并给出合理的自我解释。

177

#### SafeHome 监视的用例模板

用例：通过互联网访问摄像机监视设备—  
显示摄像机视图（ACS-DCV）。  
迭代：2。最新更改记录：V. Raman，1 月  
14 日。

主参与者：房主。  
情境目标：从任何远程地点通过互联网查  
看遍布房间的摄像头输出。  
前提条件：必须完整配置系统；必须获得

<sup>①</sup> 在该例子中，另一个参与者——系统管理员必须配置平面设计图、安装并初始化（如分配设备编号）所有的摄像头，并且通过系统平面设计图访问和测试每个摄像头能达到的特定作用。

正确的账号和密码。

触发器：房主在远离家的时候决定查看房屋内部。

场景：

1. 房主登录 SafeHome 产品网站。
2. 房主输入他的账号。
3. 房主输入两个密码（每个都至少有 8 个字符的长度）。
4. 系统显示所有的主要功能按钮。
5. 房主从主要功能按钮中选择“监视”。
6. 房主选择“选取摄像机”。
7. 系统显示房屋的平面设计图。
8. 房主从房屋的平面设计图中选择某个摄像机的图标。
9. 房主选择“视图”按钮。
10. 系统显示一个由摄像机编号确定的视图窗口。
11. 系统在视图窗口中以每秒一帧的速度显示视频输出。

异常处理：

1. 账号或密码不正确或不被确认——参看用例“确认账号和密码”。
2. 没有为该系统配置监视功能——系统显示恰当的错误消息，参看用例“配置监视功能”。
3. 房主选择“查看所有摄像机的缩略视图

快照”——参看用例“查看所有摄像机的缩略视图快照”。

4. 平面设计图不可用或是还没有配置——显示恰当的错误消息，参看用例“配置平面设计图”。
5. 遇到报警条件——参看用例“遇到报警条件”。

优先级：必须在基础功能之后实现中等优先级。

何时有效：第三个增量。

使用频率：频率较低。

参与者的连接渠道：通过基于个人计算机的浏览器和互联网连接到 SafeHome 网站。

次要参与者：系统管理员，摄像机。

次要参与者的连接渠道：

1. 系统管理员：基于个人计算机的系统。
2. 摄像机：无线连接。

未解决的问题：

1. 用什么机制保护 SafeHome 产品的雇员在未授权的情况下能使用该功能？
2. 足够安全吗？黑客入侵该功能将使最主要的个人隐私受侵。
3. 在给定摄像机视图所要求的带宽下，可以接受通过互联网的系统响应吗？
4. 若可以使用高带宽的连接，能开发出比每秒一帧更快的视频速度吗？

178

在很多情况下，不需要创建图形化表示的用户场景。然而，在场景比较复杂时，图表化的表示更有助于理解。正如我们在本书前面所提到的，UML 的确提供了图形化表现用例的能力。图 9-4 为 SafeHome 产品描述了一个初步的用例图，每个用例由一个椭圆表示。本节仅详细讨论了 ACS-DCV。

每种建模注释方法都有其局限性，用例方法也不例外。和其他描述形式一样，用例的好坏取决于它的描述者。如果描述不清晰，用例可能会误导或有歧义。用例关注功能和行为需求，一般不适用于非功能需求。对于必须特别详细和精准的需求建模情境（例如安全关键系统），用例方法就不够用了。

然而，软件工程师遇到的绝大多数情境都适用基于场景建模。如果开发得当，用例作为一个建模工具将带来很多益处。

**网络资源** 什么时候结束用例编写？关于该话题有价值的论述可以参阅 [ootips.org/use-cases-done.html](http://ootips.org/use-cases-done.html)。

### 9.3 补充用例的 UML 模型

很多基于文本的需求建模情景（即使和用例一样简单）不能简明扼要地传递信息。在这种情况下，你应能从大量的 UML 图形模型中进行选择。

#### 9.3.1 开发活动图

UML 活动图在特定场景内通过提供迭代流的图形化表示来补充用例。类似于流程图，活动图使用两端为半圆形的矩形表示一个特定的系统功能，箭头表示通过系统的流，菱形表示分支（标记从菱形发出的每个箭头），实水平线意味着并行发生的活动。ACS-DCV 用例的活动图如图 9-5 所示。应注意到活动图增加了额外的细节，而这些细节是用例不能直接描述的（隐含的）。例如，用户可以尝试有限次数地输入账号和密码，这可以通过“提示重新输入”的判定菱形来体现。

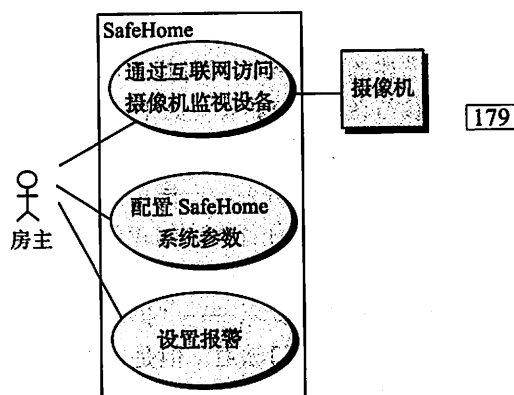


图 9-4 SafeHome 系统的初步用例图

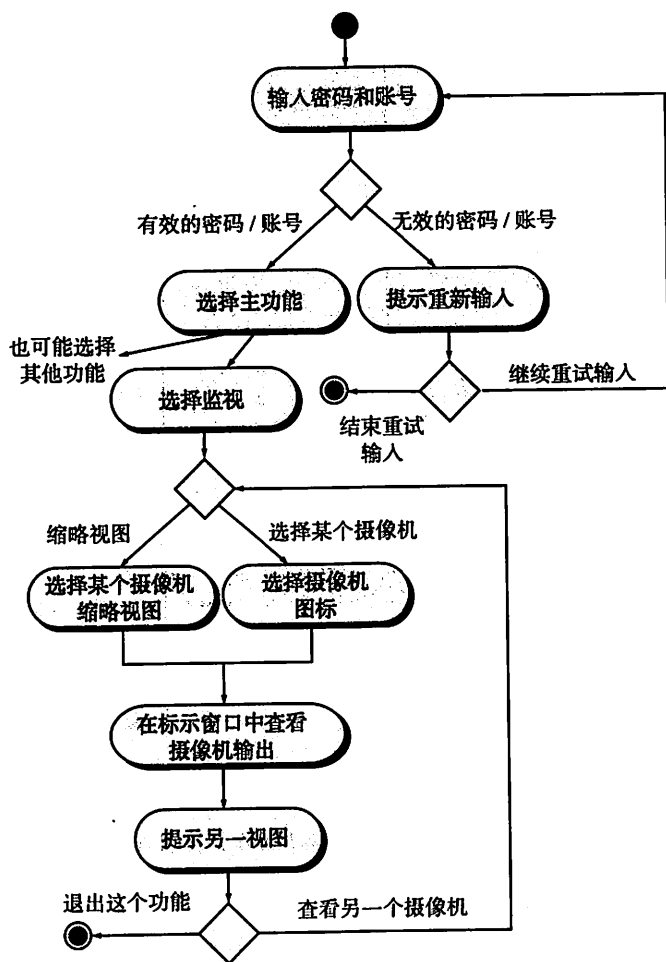


图 9-5 通过互联网访问摄像机监视设备并显示摄像机视图功能的活动图

### 9.3.2 泳道图

UML 泳道图是活动图的一种有用的变形，允许建模人员表示用例所描述的活动流，同时指出哪个参与者（如果在某个特定用例中涉及了多个参与者）或分析类（第 10 章）负责由活动矩形所描述的活动。职责由纵向分割图中的并行条表示，就像游泳池中的泳道。

**关键点** UML 泳道图表现了活动流和一些判定，并指明由哪个参与者实施。

181

三种分析类——房主、摄像机和接口——对于图 9-5 所表示的活动图中的情景具有直接或间接的责任。参看图 9-6，重新排列活动图，和某个特殊分析类相关的活动按类落入相应的泳道中。例如，接口类表示房主可见的用户接口。活动图标记出对接口负责的两个提示——“提示重新输入”和“提示另一视图”。这些提示以及与此相关的判定都落入了接口泳道。但是，从该泳道发出的箭头返回到房主泳道，这是因为房主的活动在房主泳道中发生。

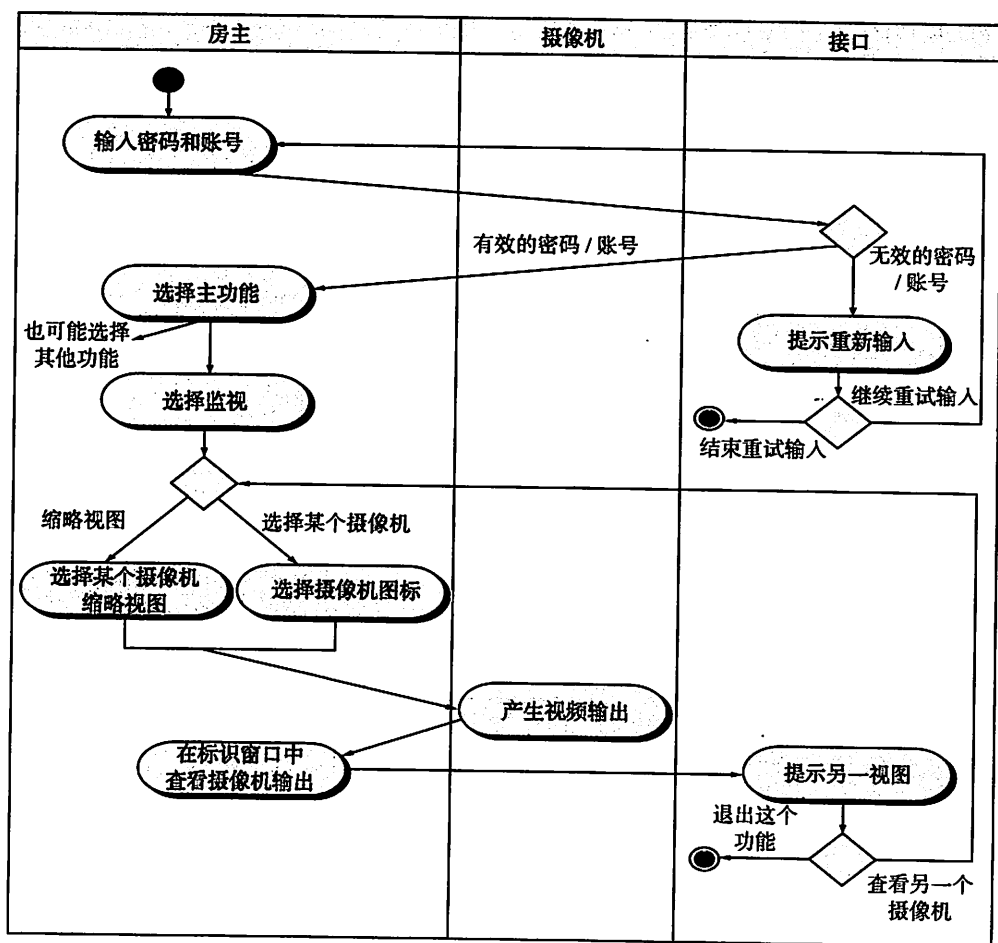


图 9-6 通过互联网访问摄像机监视设备并显示摄像机视图功能的泳道图

借助活动图和泳道图，面向过程的用例表示出各种参与者行使的一些特定功能（或其他处理步骤），以便满足系统需求。但是需求的过程视图仅表示系统的单一维度，在第 10 章和第 11 章，我们将考察需求建模的其他维度。

**引述** 好模型引导你思考，而坏模型会扭曲它。

Brian Marick

## 9.4 小结

需求建模的目标是创建各种表现形式，用其描述什么是客户需求，为生成软件设计建立基础，一旦软件建立，这些需求将用于验证。需求模型在系统级表示层和软件设计之间构造了桥梁。系统表示层描述了整个系统和业务功能，软件设计描述了软件应用的架构、用户接口和组件级的结构。

基于场景的模型从用户的角度描述软件需求。用例是主要的建模元素，它以叙述方式或以模板驱动方式描述了参与者和软件之间的交互活动。在需求获取过程中得到的用例定义了特定功能或交互活动的关键步骤。用例的形式化和详细程度各不相同，但其最终结果为所有的其他分析建模活动提供了必需的输入。还可以使用活动图说明场景，即一种类似于流程图的图形表现形式，描述在特定场景中的处理流。泳道图显示了如何给不同的参与者或类分配处理流。

## 习题与思考题

- 9.1 有没有可能在分析模型创建后立即开始编码？解释你的答案，然后说服反方。
- 9.2 一个单凭经验的分析原则指出模型“应该关注在问题域或业务域中可见的需求”。在这些域中哪些类型的需求是不可见的？提供一些例子。
- 9.3 域分析的目的是什么？如何将域分析与需求模式概念相联系？
- 9.4 有没有可能不完成图 9-3 所示的四种元素就开发出一个有效的分析模型？解释一下。
- 9.5 某个大城市的公共工程部决定开发基于 Web 的路面坑洼跟踪和修补系统 (PHTRS)。说明如下：

市民可以登录 Web 站点报告路面坑洼的地点和严重程度。上报后，该信息将记入“路面坑洼跟踪和修补系统”，分配一个标识号，保存如下信息：街道地址、大小（比例从 1 到 10）、位置（中央、路边等）、地区（由街道地址确定）以及修补优先级（由坑洼大小确定）。工作订单数据和每个坑洼有关联，数据包含坑洼位置和大小、维修组标识号、维修组内人员数量、分配的设备、修复耗时、坑洼状态（正在处理中、已修复、临时修复、未修复）、使用的填充材料数量以及修复成本（根据修复耗时、人员数量、材料和使用的设备计算）。最后，生成损失文件以便保存该坑洼所造成的损失报告信息，并包含公民的姓名、地址、电话号码、损失类型、损失金额。PHTRS 基于在线系统，可交互地进行所有查询。

为 PHTRS 系统画出 UML 用例图，你必须对用户和系统的交互方式做一些假设。

- 9.6 编写 2 ~ 3 个用例描述 PHTRS 系统中的各种参与者的角色。
- 9.7 为 PHTRS 系统的某个部分开发一个活动图。
- 9.8 为 PHTRS 系统的一个或多个部分开发一个泳道图。

## 扩展阅读与信息资源

用例是为所有需求建模方法服务的基础。讨论这一主题的书很多，包括：Gomaa (《Software Modeling: UML, Use Case, Patterns, and Architecture》, Cambridge University Press, 2011), Rosenberg 和 Stephens (《Use Case Driven Object Modeling with UML: Theory and Practice》, Apress, 2007), Denny (《Succeeding with Use Cases: Working Smart to Deliver Quality》, Addison-Wesley, 2005), Alexander 和 Maiden (eds.) (《Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle》, Wiley, 2004), Bittner 和 Spence (《Use Case Modeling》, Addison-Wesley, 2002), Cockburn [Coc01b]。其他参考资料请关注第 8 章。

UML 建模技术可以应用于分析和设计，讨论这方面的书包括：Dennis 和他的同事 (《Systems



Analysis and Design with UML Version 2.0》, 4th ed., Wiley, 2012), O'Docherty (《Object-Oriented Analysis and Design: Understanding System Development with UML2.0》, Wiley, 2005), Arlow 和 Neustadt (《UML 2 and the Unified Process》, 2nd ed., Addison-Wesley, 2005), Roques(《UML in Practice》, Wiley, 2004), Larman(《Applying UML and Patterns》, 2nd ed., Prentice-Hall, 2001), Rosenberg 和 Scott (《Use Case Driven Object Modeling with UML》, Addison-Wesley, 1999)。

关于需求的书包括 Robertson(《Mastering the Requirements Process: Getting Requirements Right》, 3rd ed., Addison-Wesley, 2012), Hull、Jackson 和 Dick (《Requirements Engineering》, 3rd ed., Springer, 2010), Alexander 和 Beus Dukic(《Discovering Requirements: How to Specify Products and Services》, Wiley, 2009)。

网上有很多关于需求建模的信息。可以参考 SEPA 网站 [www.mhhe.com/pressman](http://www.mhhe.com/pressman) 上有关分析建模的最新参考文献。