

第3章 需求建模：场景、信息与类分析

在技术层面上,软件工程开始于一系列建模工作,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,是系统的第一个技术表示。而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,建立生成软件设计的基础。一旦软件建立,就能定义一组可被确认的需求。需求模型在系统级表示层和软件设计之间构造了桥梁。系统表示层描述整个系统和业务功能,软件设计描述软件应用的架构、用户接口和组件级的结构。

3.1 需求分析

需求分析产生所开发软件的规格说明,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,并且让软件工程师(有时也称软件分析师或软件架构师)细化在前期需求工程的起始、导出、谈判任务中建立的基础需求。

需求建模活动产生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。
- 数据模型:描述问题信息域的模型。
- 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。
- 面向流程的模型:表示系统的功能元素,并且描述功能元素在系统中运行时怎样进行数据变换的模型。
- 行为模型:描述如何将软件行为看做是外部“事件”后续的模型。

其中,基于场景的建模技术在软件工程界迅速发展;数据建模是较为特殊的技术,适用于当一个应用问题必须生成或处理一个复杂的信息空间时;面向类建模表示面向对象类和允许的系统功能间的有效协作。

这些模型为软件设计者提供信息,这些信息可以转化为结构、接口和构件级的设计。最终,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。

3.1.1 总体目标和原理

在需求建模过程中,软件工程师主要关注“做什么”而不是“怎么做”。包括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。

由于在初期就要得到完整的需求规格说明是不可能的,因此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。

需求模型必须实现 3 个主要目标:

① 描述客户需要什么;

② 为软件设计奠定基础;

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之间建立了桥梁。这里,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,其关系如图 3-1 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。通常很难清楚地区分这两个重要的建模活动之间的设计和分析工作,有些设计会作为分析的一部分进行,而有些分析将在设计中进行。

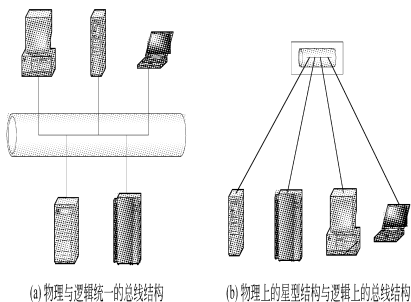


图 3-1 分析模型在系统描述和设计模型之间建立桥梁

3.1.2 分析的经验原则

创建分析模型时,应该遵循的一些有价值的经验原则如下。

- 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”,即不要试图解释系统将如何工作。
- 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迟到设计阶段再考虑。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,才考虑实现数据库所必需的类、访问数据库的功能,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,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,利益相关的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,质量管理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,不要添加附加图表;如果一个简单列表就够用,就不要使用复杂的表示方法。

3.1.3 域分析

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。这将有可能会使产品投放市场的时间提前,并减少开发费用。

对域分析的描述可以是“软件域分析是识别、分析和详细说明某个特定应用领域的公共需求,特别是那些在该应用领域内被多个项目重复使用的需求……面向对象的域分析是在某个特定应用领域内,根据通用的对象、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,就是查找或创建那些广泛应用的分析类和(或)分析模式,使其能够复用。

域分析可以被看做是软件过程的一个普适性活动,而不与任何一个软件项目相关。域分析师的角色类似于机械制造业中一名优秀的刀具工。刀具工的工作是设计并制造工具,这些工具可以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,以便用于类似但不一定完全相同的应用。

图 3-2 所示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。



图 3-2 域分析的输入和输出

3.1.4 需求建模的方法

所谓结构化分析,是一种考虑数据和处理的需求建模方法,其中的处理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数据对象在系统内流动时处理如何转换数据。需求建模的第二种方法称做面向对象的分析,这种方法关注定义类和影响客户需求的类之间的协作方式。UML 和统一过程主要是面向对象的分析方法。问题是怎么组合这些表示手段,才能为利益相关者提供最好的软件需求模型和过渡到软件设计的最有效方法。

如图 3-3 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。基于类的元素建模于系统操作的对象,应用在这些对象间影响操作和对象间关系(某层级)的操作,以及定义的类型间发生的协作。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。

需求建模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。软件团队必须想办法保持模型的简单性,只使用那些为模型增加价值的建模元素。

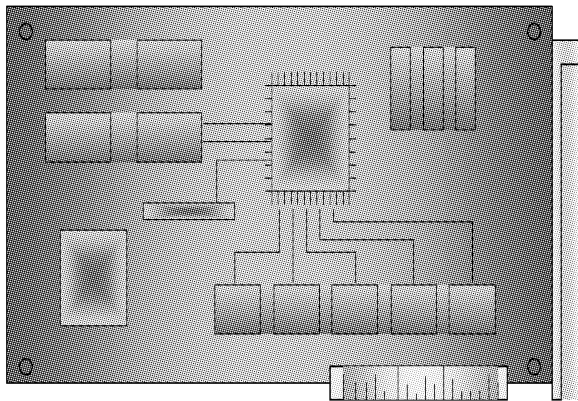


图 3-3 需求模型的元素

3.2 基于场景建模

尽管度量基于计算机的系统或产品的成果可以有多种方式,但用户满意度仍然是其中最重要的。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,软件团队就能够更好地、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 需求建模将从开发用例、活动图和泳道图形式的场景开始。

3.2.1 新建初始用例

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,但需要知道:

- ① 编写什么;
- ② 应该有多详细;
- ③ 如何组织说明,想让用例像一个需求建模工具那样提供价值,就必须回答这些问题。

两个首要的需求工程工作是起始和导出,它们提供了开始编写用例所需要的信息。运用需求收集会议、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。通过与利益相关者交流,可以从所需系统功能的列表或评估活动图(作为需求建模中的一部分而开发)获得。

3.2.2 细化初始用例

为全面理解用例描述功能,对交互操作给出描述是非常必要的。因此,主场景中的每个步骤将通过如下提问得到评估。

- 在这一状态点,参与者能进行一些其他动作吗?
- 在这一状态点,参与者有没有可能遇到错误的条件? 如果有可能,这些错误会是什么?
- 在这一状态点,参与者有没有可能遇到其他行为(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(Secondary Scenarios)。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。

此外,还应该研究下面的问题。

- 在这个用例中,是否有某些具有“有效功能”的用例出现? 包括引用确认功能,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,一个用户动作是等待一个应答,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一个出错条件。

这里列出的扩展部分是由具有因果关系的提问和回答问题开发的。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,就应该在检测出后马上处理这个条件。在某些情况下,异常处理可能拖累影响其他用例处理条件的开发。

3.2.3 编写正规用例

当一个用例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,应采用更正规的方法。

在很多情况下,不需要创建图形化表示的用户场景。然而,当场景比较复杂时,图形化表示将更有助于理解。UML 提供了图形化表现用例的能力。图 3-4 是为 SafeHome 产品描述的一个初步的用例图,每个用例由一个椭圆表示。

每种建模注释方法都有局限性,如果描述不清晰,用例也可能误导或有歧义。一个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景(例如安全关键系

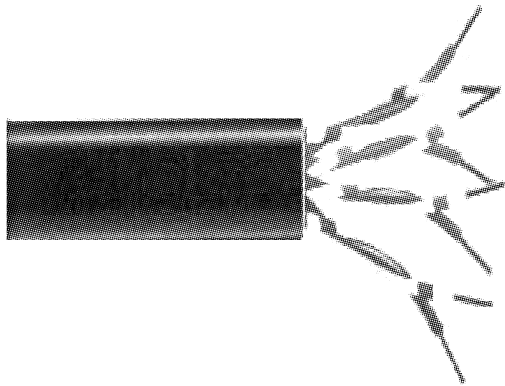


图 3-4 SafeHome 系统的初步用例图

统),用例方法就不够用了。但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,用例是重要的建模工具之一。

3.3 补充用例的 UML 模型

很多基于文本的需求建模情景甚至和用例一样简单,无法简明扼要地传递信息。在这种情况下,可以从 UML 图形模型中选择。

3.3.1 开发活动图

在特定场景内,UML 活动图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,判定菱形表示判定分支(标记从菱形发出的每个箭头),实心水平线意味着并行发生的活动,如图 3-5 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(是隐含的)。例如,用户尽可以尝试有限次数地输入用户身份证号(ID)和密码。这可以通过“提示重新输入”的判定菱形来体现。

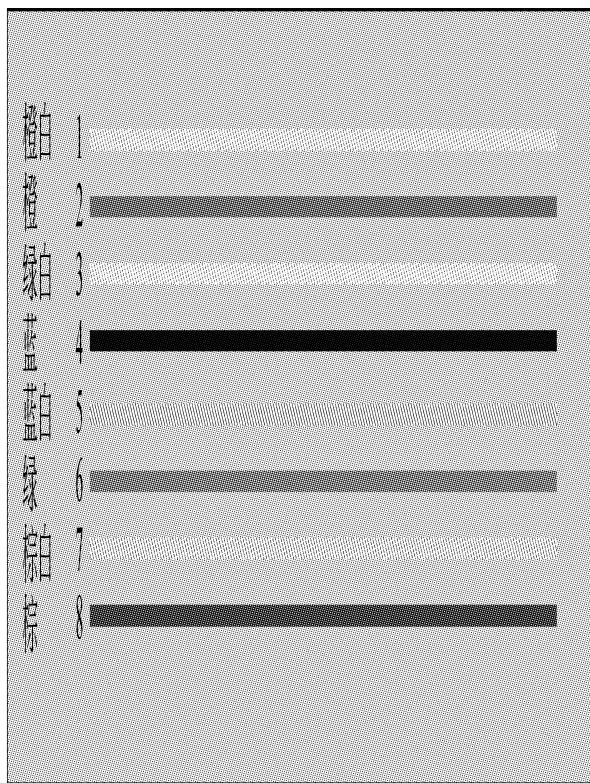


图 3-5 通过因特网进入摄像机监视并显示摄像机视图功能的活动图

3.3.2 泳道图

UML 泳道图是活动图的一种有用变形,可以让建模人员表示用例所描述的活动流,同时指示哪个参与者(如果在某个特定用例中涉及了多个参与者)或分析类是由活动矩形所描述的活动来负责。职责由纵向分割图的并行条表示,就像游泳池中的泳道。

3 种分析类(房主、摄像机和接口)对图 3-5 表示的活动图中的情景具有直接或间接的责任。参看图 3-6,重新排列活动图,与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。

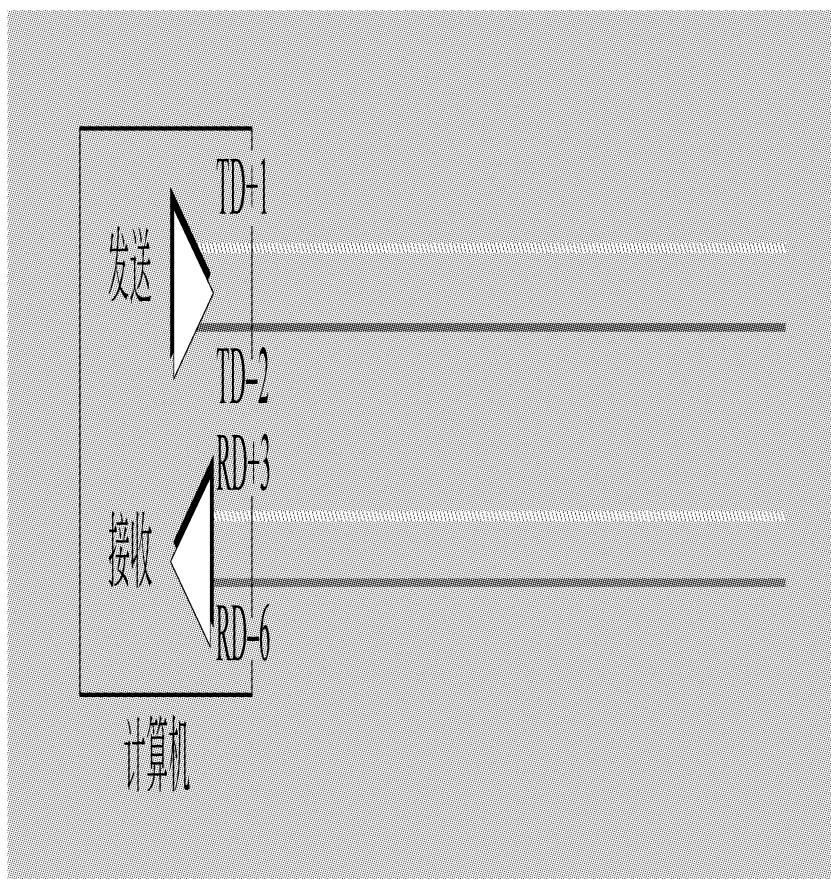


图 3-6 通过因特网进入摄像机监视并显示摄像机视图功能的泳道图

伴随着活动图和泳道图,面向过程的用例表示各种参与者行使一些特定功能(或其他处理步骤),以便满足系统需求。

3.4 数据建模概念

如果软件需求包括建立、扩展需求,或者具有数据库的接口,或者必须构建和操作复杂的数据结构,软件团队可以选择建立一个数据模型,作为需求建模的一部分。软件工程师或分析师需要定义在系统内处理的所有数据对象,数据对象之间的关联以及其他与此相关的信息。实体关系图(ERD)描述了这些问题,并提供了在一个应用项目中输入、存储、转换和产生的所有数据对象。

3.4.1 数据对象

数据对象必须由软件理解的复合信息表示。所谓复合信息,是指具有若干不同的特征或属性的事物。因此,“宽度”(单个的值)不是有效的数据对象,但是“维度”(包括宽度、高度和深度)可以被定义为一个对象。

数据对象可能是外部实体(例如产生或使用信息的任何东西)、事物(例如报告或显示)、偶发事件(例如电话呼叫)或事件(例如警报)、角色(例如销售人员)、组织单位(例如财务部)、地点(例如仓库)或结构(例如文件)。例如,一个人或一部车可以被认为是数据对象,在某种意义上,它们可以用一组属性来定义。数据对象描述包括了数据对象及其所有属性。

数据对象只封装数据(没有操作数据的引用。注意将数据对象与面向对象方法中的“类”或“对象”区分开来)。因此,数据可以表示为如图 3-7 所示的一张表,表头反映了对象的属性。在这个例子中,汽车是通过生产商(Make)、车型(Model)、标识号(ID#)、车体类型(Body type)、颜色(Color)和车主(Owner)定义的。该表的主体表示了数据对象的特定实例。例如,Chevy Corvette 牌的车是数据对象“汽车”的一个实例。

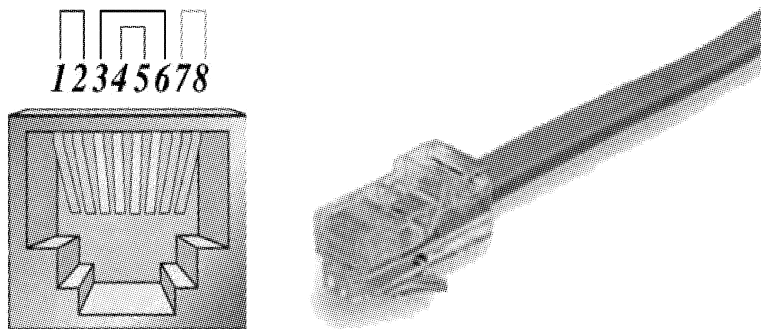


图 3-7 数据对象的表格表示

3.4.2 数据属性

数据属性定义了数据对象的性质,作用如下。

① 为数据对象的实例命名；

② 描述这个实例；

③ 建立对另一个表中的另一个实例的引用。另外，必须把一个或多个属性定义为标识符。也就是说，当要找到数据对象的一个实例时，标识符属性成为一个“键”。在某些情况下，标识符的值是唯一的，但不是必需的。在数据对象“汽车”的例子中，标识号是一个合理的标识符。

对问题环境的理解，可以恰当地确定特定数据对象的一组属性。“汽车”的属性可以用于汽车运输部门的应用系统，但这些属性对于汽车制造公司来说是无用的，他们需要制造中的控制软件。在后一种情况下，“汽车”的属性可能也包括标识号、车体类型和颜色，但为了使汽车在制造的控制环境下成为一个有用的对象，必须增加许多其他的属性，如内部代码、驱动系统类型、车内包装设计师、传动类型等。

3.4.3 关系

数据对象可以以多种不同的方式与另一个数据对象连接。考虑以下两个数据对象：

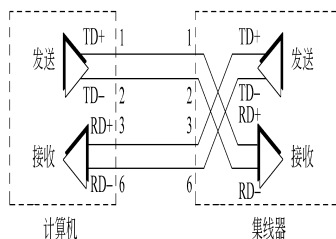


图 3-8 数据对象之间的关联关系

person 和 car。这些对象可以使用图 3-8(a)所示的简单标记表示。person 和 car 之间可以建立联系，因为这两个对象之间是相关的。必须理解将要构建的软件环境中人（这里是指车主）和车的角色。可以用一组“对象/关系对”来定义相互关系，例如：拥有车的人和汽车投保人。关系“拥有”和“驾车投保”定义了 person 和 car 之间的相关连接。图 3-8(b)以图形方式说明了这些对象/关系对，图中标注的箭头提供了关联方向信息，这一方向信息通常可以减少歧义或误解。

3.5 基于类的建模

基于类的建模表示了系统操作的对象、应用于对象间能有效控制的操作（也称为方法或服务）、这些对象间（某种层级）的关系以及已定义类之间的协作。基于类的分析模型元素包括类和对象、属性、操作、类的职责协作者（CRC）模型、协作图和包。

3.5.1 识别分析类

环顾房间，就可以发现一组容易识别、分类和定义（就属性和操作而言）的物理对象。但当“环顾”软件应用的问题空间时，了解类和对象就没有那么容易了。

通过检查需求模型开发的使用场景，对系统开发的用例进行“语法解析”，可以进行类的识别。其中，每个名词或名词词组可以确定为类，将这些名词输入到一个简单的表中，标注出同义词。如果要求某个类（名词）实现一个解决方案，这个类就是解决方案空间的

一部分；否则，如果只要求某个类描述一个解决方案，这个类就是问题空间的一部分。分析类表现为如下方式之一。

- 外部实体(如其他系统、设备、人员)，产生或使用基于计算机系统的信息。
- 事物(如报告显示、字母、信号)，是问题信息域的一部分。
- 偶发事件或事件(如所有权转移或完成机器人的一组动作)，在系统操作环境中发生。
- 角色(如经理、工程师、销售人员)，由和系统交互的人员扮演。
- 组织单元(如部门、组、团队)，和某个应用系统相关。
- 场地(如制造车间或码头)，建立问题的环境和系统的整体功能。
- 结构(如传感器、四轮交通工具、计算机)，定义了对象的类或与对象相关的类。

这种分类只是文献中提出的大量分类之一(另一个重要的分类是指定义实体、边界和控制类)。例如，Budd 建议了一种类的分类法，包括数据产生者(源点)和数据使用者(汇点)、数据管理者、查看或观察者类以及帮助类。还需要特别注意什么不能是类或对象。

通常，不应该使用“命令过程式的名称”为类命名。例如，如果医疗图像系统的软件开发人员使用名字“`InvertImage`”甚至用“`ImageInversion`”定义对象，就可能犯下一个小小的错误。从软件获得的 `Image` 当然可能是一个类(这是信息域中的一部分)，图像的翻转是适用于该对象的一个操作，程序很可能将翻转定义为对于对象 `Image` 的一个操作，但是不可能定义单独的类来暗示“图像翻转”。面向对象的目的是封装，但仍保持独立的数据以及对数据的操作。

在分析模型中，分析师可以考虑每个潜在类是否应该使用如下这些特征。

(1) 保留信息。只有记录潜在类的信息才能保证系统正常工作，这种分析过程中的潜在类是有用的。

(2) 所需服务。潜在类必须有一组可确认的操作，这组操作能用某种方式改变类的属性值。

(3) 多个属性。在需求分析过程中，焦点应在“主”信息上。事实上，只有一个属性的类可能在设计中有用，但在分析活动阶段，最好把它作为另一个类的某个属性。

(4) 公共属性。可以为潜在类定义一组属性，这些属性适用于类的所有实例。

(5) 公共操作。可以为潜在类定义一组操作，这些操作适用于类的所有实例。

(6) 必要需求。在问题空间中出现的外部实体和任何系统解决方案运行时所必需的生产或消费信息，几乎都被定义为需求模型中的类。

3.5.2 描述属性

属性描述了已经包含在需求模型中的类。实质上，属性是定义类，以澄清类在问题空间的环境下意味着什么。例如，如果建立一个系统来跟踪职业棒球手的统计信息，类 `Player` 的属性与用于职业棒球手养老系统中的属性是截然不同的。前者中，属性是与名字、位置、平均击球次数、担任防守百分比、从业年限、比赛次数等相关的。后者中，某些属性具有相同的含义，另外一些属性将被替换，例如平均工资、享受优惠权后的信用、所选的

养老计划、邮件地址等。

为了给分析类开发一个有意义的属性集合,软件工程师应该研究用例,并选择那些合理“属于”类的“事物”。此外,每个类都应回答如下问题:什么数据项(组合项和/或者基本项)能够在当前问题环境内完整地定义这个类?

3.5.3 定义操作

操作定义了某个对象的行为。尽管存在很多不同类型的操作,但通常可以粗略地分为4种类型:

- ① 以某种方式操作数据(如添加、删除、重新格式化、选择);
- ② 执行计算的操作;
- ③ 请求某个对象状态的操作;
- ④ 监视某个对象发生某个控制事件的操作。这些功能通过在属性和/或相关属性上的操作实现。因此,操作必须“理解”为类的属性和相关属性的性质。

第一次迭代要导出一组分析类的操作时,可以再次研究处理叙述(或用例),并合理地选择属于该类的操作,为此可以再次进行语法解析,并分离动词。这些动词中的一部分将是合法的操作,并能够很容易连接到某个特定类。

另外,对于语法分析,分析师能通过考虑对象间发生的通信获得对其他操作更为深入的了解。对象通过传递信息与另一个对象通信。在继续对操作进行说明之前,可以探测到更翔实的信息。

3.5.4 类—职责—协作者建模

类—职责—协作者(Class-Responsibility-Collaborator,CRC)建模提供了一个简单方法,可以识别和组织与系统或产品需求相关的类。CRC模型实际上是表示类的标准索引卡片的集合。这些卡片分3部分,顶部写类名,卡片主体左侧部分列出类的职责,右侧部分列出类的协作者。

事实上,CRC模型可以使用真的或虚拟的索引卡,意图是开发有组织表示的类。职责是和类相关的属性和操作。简单地说,职责就是“类所知道或能做的任何事”。协作者提供完成某个职责所需要信息的类。通常,协作意味着信息请求或某个动作请求。

FloorPlan类的一个简单CRC索引卡如图3-9所示。CRC卡上列出的职责只是初步的,可以添加或修改。职责栏右边的Wall和Camera是需要协作的类。

1. 类

类的分类可以通过如下分类方式进行扩展。

(1) 实体类,也称做模型或业务类,是从问题说明中直接提取出来的(如FloorPlan和Sensor)。这些类一般代表保存在数据库中 and 贯穿于应用程序(除非被明确删除)的事物。

(2) 边界类,用于创建用户可见的和在使用软件时交互的接口(如交互屏幕或打印的报表)。实体类包含对用户来说很重要的信息,但并不显示这些信息。设计边界类的职责是管理实体对象对用户的表示方式。例如,一个称做 Camera Window 的边界类负责显示 SafeHome 系统监视摄像机的输出。

(3) 控制类自始至终管理着“工作单元”。也就是说,设计控制类可以管理如下内容:

- ① 实体类的创建或更新;
- ② 当边界类从实体对象获取信息后的实例化;
- ③ 对象集合间的复杂通信;
- ④ 对象间或用户和应用系统间交换数据的确认。通常直到设计开始时才开始考虑控制类。

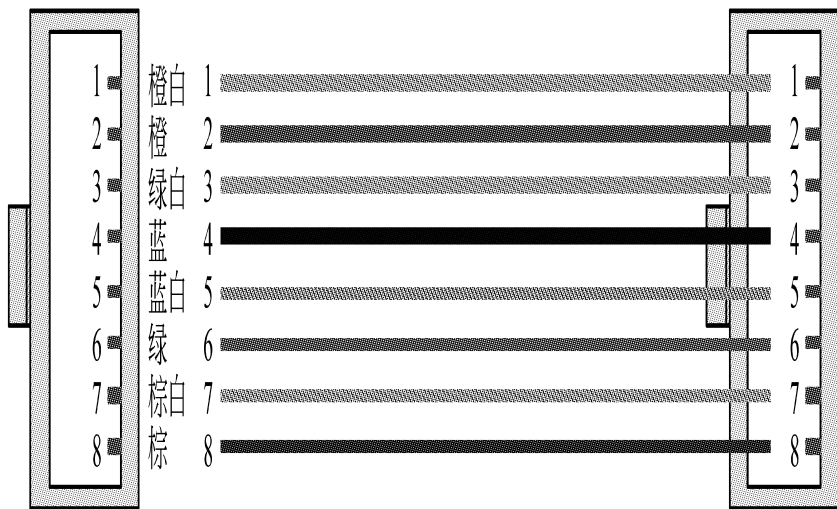


图 3-9 CRC 模型索引卡

2. 职责

给类分配职责时,建议考虑以下 5 个指导原则。

(1) 智能系统应分布在所有类中,以最好地满足问题的需求。每个应用系统都包含一定程度的智能,也就是系统内含有它知道的以及所能完成的。智能在类中可以有多种分布方式,建模时可以把“不灵巧”类(几乎没有职责的类)作为一些“灵巧”类(有很多职责的类)的从属。尽管该方法使系统中的控制流简单易懂,但它把所有的智能集中在少数类,使得变更更为困难;将会需要更多的类,因此需要更多的开发工作。

如果智能系统更平均地分布在应用系统的所有类中,每个对象只了解和执行一些事情(通常是适度集中),系统的内聚性提高,将提高软件的可维护性,并减少变更的副作用。

为了确定是否恰当地分布了智能系统,应该评估每个 CRC 模型索引卡上标记的职责,以确定某个类是否应该具有超长的职责列表。职责列表超长,就表明智能太集中(可

能需要将一个类分成多个类或子系统,以便更有效地分布智能)。此外,每个类的职责应表现在同一抽象层上。例如,在聚合类 `CheckingAccount` 操作列表中,评审人员注意到 2 项职责:账户余额和已结算的支票。第 1 个操作的职责意味着复杂的算术和逻辑过程;第 2 个操作的职责是指简单的办事员活动。既然这两个操作不是相同的抽象级别,已结算的支票应该放在 `CheckEntry` 的职责中,这是由聚合类 `CheckingAccount` 压缩得到的一个类。

(2) 每个职责的说明应尽可能具有普遍性。这条指导原则意味着应在类的层级结构的上层保持职责(属性和操作)的通用性(因为它们更有一般性,将适用于所有子类)。

(3) 信息和与之相关的行为应放在同一个类中。这实现了面向对象原则中的封装,数据和操作数据的处理应包装在一个内聚单元中。

(4) 某个事物的信息应局限于一个类中,而不要分布在多个类中。通常应由一个单独的类负责保存和操作某特定类型的信息,这个职责不应由多个类分担。如果信息是分布的,软件将变得更难维护,测试也会面临更多挑战。

(5) 适合时,职责应由相关类共享。很多情况下,各种相关对象必须在同一时间展示同样的行为。例如,考虑一个视频游戏,必须显示如下类: `Player`、`PlayerBody`、`PlayerArms`、`PlayerLegs` 和 `PlayerHead`,每个类都有各自的属性(如 `position`、`orientation`、`color` 和 `speed`),并且所有这些属性都必须在用户操纵游戏杆时更新和显示。因此,每个对象必须共享职责 `update()` 和 `display()`。`Player` 知道什么时候发生了某些变化,并且需要 `update()` 操作,它和其他对象协作,获得新的位置或方向,但是每个对象控制各自的显示。

3. 协作

类有一种或两种方法实现职责:①类可以使用其自身的操作控制各自的属性,实现特定的职责;②一个类可以和其他类协作。

协作是从客户职责实现的角度表现从客户到服务器的请求。协作是客户和服务端之间契约的具体实现。如果为了实现某个职责,需要发送任何消息给另一个对象,我们就说这个对象和其他对象有协作。单独的协作是单向流,即表示从客户到服务器的请求。从客户的角度看,每个协作都和服务器的某个特定职责实现相关。

要识别协作,可以通过确认类本身是否能够实现自身的每个职责。如果不能实现每个职责,就需要和其他类交互,因此就要有协作。

为帮助识别协作者,分析师可以检查类之间 3 种不同的通用关系:① `is-part-of` (是……一部分)关系;② `has-knowledge-of` (有……的知识)关系;③ `depends-upon` (依赖)关系。下面将简单地说明这 3 种通用关系。

属于某个聚合类一部分的所有类可通过 `is-part-of` 关系和聚合类连接。考虑前面提到的视频游戏中定义的类,`PlayerBody` 是 `Player` 的一部分,`PlayerArms`、`PlayerLegs` 和 `PlayerHead` 也类似。在 UML 中,使用如图 3-10 所示的聚合方式表示这些关系。

当一个类必须从另一个类中获取信息时,就建立了 `has-knowledge-of` 关系。

在所有情况下,把协作类的名称记录在 CRC 模型索引卡上,紧靠在协作的职责旁边。

因此,索引卡包含一个职责列表以及相关的能够实现这些职责的协作,如图 3-9 所示。

当开发出一个完整的 CRC 模型时,利益相关者可以使用如下方法评审模型。

(1) 所有参加(CRC 模型)评审的人员拿到一部分 CRC 模型索引卡。拆分协作卡片(也就是使每个评审员不得有 2 张存在协作关系的卡片)。

(2) 分类管理所有的用例场景(以及相关的用例图)。

(3) 评审组长细致地阅读用例。当评审组长看到一个已命名的对象时,给拥有相应类索引卡的人员一个令牌。

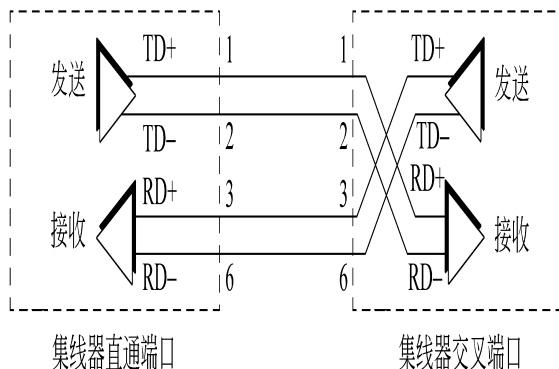


图 3-10 复合聚合类

3.5.5 关联和依赖

在很多例子中,两个分析类以某种方式相互联系着,就如同彼此相互联系的两个数据对象。在 UML 中,这些联系被称做关联(associations)。在某些情况下,关联可以更进一步地指出多样性,其中“一个或多个”使用 1.. 表示,“0 个或多个”使用 0.. 表示。在 UML 中,星号表示范围无上界。其他的多样性关联(一对一、一对多、多对多、一对某指定上下限的范围等)可以标识为关联的一部分。

在很多事例中,两个分析类之间存在客户-服务器关系。在这种情况下,客户类以某种方式依赖于服务器类,并且建立了依赖关系。依赖是由一个构造型(stereotype)定义的。在 UML 中,构造型是一个“可扩展机制”,允许软件工程师定义特殊的建模元素,这些建模元素的语义是由用户自定义的。在 UML 中,构造型由一对尖括号表示(如 <<stereotype>>)。

3.5.6 分析包

分析建模的一部分重要工作是分类,也就是将分析模型的各种元素(如用例、分析类)以一种方式分类,分组打包后称其为分析包,并取一个有代表性的名字。

3.6 习题

1. 请参考课文内容及其他资料,完成下列选择题。

(1) 某软件公司欲开发一个图像处理系统,在项目初期,开发人员对算法的效率、操作系统的兼容性和人机交互的形式等情况并不确定。在这种情况下,采用()方法比较合适。

- A. 瀑布式 B. 形式化 C. 协同开发 D. 快速原型

(2) 面向对象分析的任务不包括()。

- A. 建模系统功能 B. 发现并确定业务对象
C. 建模各对象的状态 D. 组织对象并确定对象间的关系

(3) 关于用例和类,错误的说法是()。

- A. 两者都属于模型图的构成元素
B. 存在抽象用例和抽象类
C. 类图描述系统的部分静态视图,用例图描述系统与用户之间的交互视图
D. 两者都可以用来描述系统的内部结构

(4) ()适用于描述复杂算法的执行流程。

- A. 活动图 B. 状态图 C. 类图 D. 用例图

(5) 在面向对象方法中,信息流是通过向参与者或内部对象发送消息形成的。

()用于描述进出系统的信息流。

- A. 协作图 B. 顺序图 C. 数据流图 D. 流程图

(6) 面向对象设计是模型驱动和用例驱动的,整个设计过程将(①)作为输入,并生成(②)作为输出。

- ① A. 逻辑数据流图 B. 设计文档和用户说明
C. 需求类图 D. 需求模型
② A. 物理数据流图 B. 设计文档和用户说明
C. 设计类图和系统顺序图 D. 设计模型

(7) 用例(use case)用来描述系统对事件做出响应时所采取的行动。用例之间是具有相关性的。在一个“订单输入子系统”中,创建新订单和更新订单都需要核查用户账号是否正确。用例“创建新订单”、“更新订单”与用例“核查客户账号”之间是()关系。

- A. 包含(include) B. 扩展(extend)
C. 分类(classification) D. 聚集(aggregation)

(8) 面向对象的设计模型包含以(①)表示的软件体系结构图,以(②)表示的用例实现图、完整精确的类图、针对复杂对象的状态图和用以描述流程化处理的活动图等。

- A. 部署图 B. 包图 C. 协同图 D. 交互图

(9) 下列关于不同软件开发方法所使用的模型描述中,正确的是()。

- A. 在进行结构化分析时,必须使用数据流图和软件结构图这两种模型
B. 采用面向对象开发方法时,可以使用状态图和活动图,对系统的动态行为进

行建模

C. 实体联系图(E-R图)是在数据库逻辑结构设计时才开始创建的模型

D. UML 的活动图与程序流程图的表达能力等价

(10) 净室软件工程是软件开发的一种(①)方法,可以开发较高质量的软件。它使用盒结构规约进行分析和建模,并将(②)作为发现和排除错误的主要机制,采用统计测试获取验证软件可靠性所需要的信息。

- | | | | |
|----------|--------|--------|----------|
| ① A. 形式化 | B. 迭代式 | C. 瀑布式 | D. 交互式 |
| ② A. 测试 | B. 仿真 | C. 实验 | D. 正确性验证 |

2. 请参考课文内容以及其他资料,完成下列问答题。

(1) 有没有可能在分析模式创建后立即开始编码? 解释你的答案,然后说服反方。

(2) 一个单凭经验的分析原则是: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? 请提供一些例子。

(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?

(4) 有没有可能不完成如图 3-3 所示的 4 种元素就开发出一个有效的分析模型? 解释一下。

(5) 什么是分析包? 如何使用分析包?

3.7 实验与思考：信息系统需求分析

3.7.1 实验目的

本节“实验与思考”的目的如下。

(1) 熟悉基于场景、基于数据、基于类建模的需求分析的概念和基本内容。

(2) 熟悉 UML 模型的设计与表达方法。

(3) 阅读与深入了解案例资料,尝试完成影音产品在线管理及销售系统和宏观经济数据库建设的需求分析。

3.7.2 工具/准备工作

开始本实验之前,请认真阅读课程的相关内容。

需要准备一台带有浏览器,能够访问因特网的计算机。

3.7.3 实验内容与步骤

1. 开发活动的用例

请参考教材内容,尝试为如下活动之一开发一个完整的用例。

(1) 在 ATM 机取款。

- (2) 在餐厅使用信用卡付费。
- (3) 使用一个在线经纪人账户购买股票。
- (4) 使用网络书店搜索书(某个指定主题)。
- (5) 你的指导老师指定的一个活动。

请以电子稿形式完成此作业,并将作业以压缩文件格式用电子邮件交给你的实验指导老师。

<班级>_<学号>_<姓名>_实验 3.rar (或.zip)
请记录:本次实验能够顺利完成吗?如果不能,请说明为什么。

2. 案例分析:在线销售系统需求分析

阅读以下关于需求分析的叙述,回答问题 1 至问题 3。

某软件企业为网络音像制品销售公司 W 重新开发一套影音产品在线管理及销售系统,以改进原有系统 AWMSS 中存在的问题。在系统需求分析阶段,完成的工作包括如下内容。

(1) 系统分析员老王利用 PIECES 框架^①组织了系统需要获取的非功能性需求,如表 3-1 所示。

表 3-1 PIECES 框架

非功能性需求类型	需要获取的需求
性能(Performance)	(a)
信息(Information)	(b)
经济(Economics)	(c)
控制(Control)	(d)
效益(Efficiency)	(e)
服务(Service)	(f)

(2) 项目组小赵从 W 公司客户代表处了解到,现有系统中经常有会员拒绝履行订单,并将其作为问题记录了下来。老王指出小赵并未发现系统真正的问题,并以会员拒绝履行订单为例,利用如图 3-11 所示的鱼骨图分析了系统中真正存在的问题。

(3) 获取相应需求后,将需求记录下来,形成需求定义文档,同其他项目信息合并形成需求陈述,作为需求分析阶段最终的交付成果。

^① PIECES 框架:是系统非功能性需求分类的技术,对各种类型的需求进行分类,使得类似的需求可以组织起来,达到汇报、跟踪和验证的目的,还可能帮助确定可能忽略的需求。James Wetherbe 提出的 PIECES 框架能够完整、准确、快速地确定信息系统的需求,确认业务中存在的问题、机会和改进目标。包括性能(Performance)、信息(Information)、经济(Economics)、控制(Control)、效益(Efficiency)和服务(Service)6 个类别。

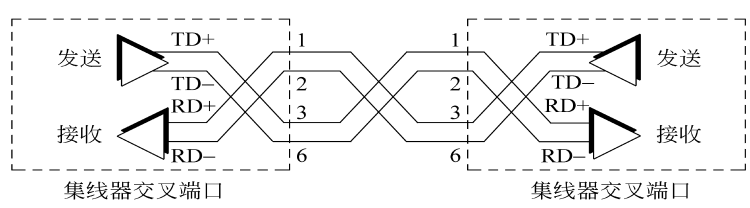


图 3-11 会员拒绝履行合同

〔问题 1〕 PIECES 框架的主要作用是什么？

答：_____

请将以下需要获取的需求(1)~(8)根据 PIECES 框架进行分类,并将序号填入表 3-1 对应的单元格内。

- (1) 系统能否采用新方法,以降低使用资源的成本？
- (2) 系统可接受的吞吐率是多少？
- (3) 系统可接受的响应时间是多少？
- (4) 应该减少多少开支或增加多少收益？
- (5) 对用户隐私有什么要求？
- (6) 对系统的可靠性和可用性有什么要求？
- (7) 系统中需要包括哪些文档和培训材料？
- (8) 对外部系统的接口是什么？

请记录：

- | | |
|-----------|-----------|
| (a) _____ | (b) _____ |
| (c) _____ | (d) _____ |
| (e) _____ | (f) _____ |

〔问题 2〕 请将下列问题按照不同的类型序号填入图 3-11 所示的鱼骨图(g)~(n)中。

- (1) 缺少强制履行合同的规定。
- (2) 合同相关信息没有通知到会员。
- (3) 没有催单提示客户。
- (4) 没有跟踪执行情况。
- (5) 设备成本太高,造成价格不合理。
- (6) 合同的履行缺乏灵活性。
- (7) 账务问题或者隐瞒相关内容。
- (8) 价格太高并且无法修改。

请记录：

- | | |
|-----------|-----------|
| (g) _____ | (h) _____ |
| (i) _____ | (j) _____ |
| (k) _____ | (l) _____ |

(m) _____

(n) _____

[问题 3] 一份需求定义文档应该包括哪些内容? 对于与系统开发相关的人员,即系统所有者、用户、系统分析人员、设计人员和构造人员、项目经理,需求定义文档各有什么作用?

答: _____

3. 案例分析: 宏观经济数据库建设

阅读以下关于宏观经济数据库建设的叙述,回答问题 1 至问题 3。

A 市经过软课题研究,已经形成了一整套宏观经济指标体系,用于描述该市的经济状态,涉及包括区域 GDP、人口与就业、城市建设与投资、财政收入和支出、土地、进出口贸易、社会保障、人民生活、制造业和高新技术产业等方面,并为主要指标建立了计算模型。与宏观经济指标有关的数据称为宏观经济数据,它广泛地分布于政府统计部门、计划部门、财政部门、税收部门、教育部门、商业部门、物价部门、农业主管部门、工业信息化主管部门等,还分布于金融部门、大中型企业等。这些部门针对自身业务,多数都有自己的管理信息系统或办公自动化系统,主要的宏观经济数据都已经实现了电子化。

目前,A 市宏观经济指标的获得还采取传统方式,即通过有关部门逐级报表汇总宏观经济数据的方式和统计调查的方式得到。统计调查方式存在系统性误差,而报表汇总方式则存在基础数据不准确、人为影响较大、指标分析灵活性差等问题。为准确掌握经济动态信息,进一步规范各级经济信息资源管理行为,加快信息资源共享,提高政府的管理科学化和服务社会化水平,该市决定建设宏观经济数据库,将宏观经济数据统一管理。

[问题 1] 为稳步推动 A 市宏观经济数据库建设,A 市市政府委托咨询公司 B 进行项目原型研究。B 公司经过调研,认为 A 市电子政务网络、存储和计算平台非常完善,为宏观经济数据库的建设奠定了良好的基础。请用 200 字以内文字指出此时 B 公司进行原型分析时应重点做好哪几方面的工作?

答: _____

[问题 2] 经过分析比较,B 公司确定了“物理分散,逻辑集中”的建设思路,因此未来宏观经济数据库是一个分布式数据库系统。请用 200 字以内文字简要分析,除了数据交换机制外,是否还需要建设一个集中的数据库? 并叙述理由。

答: _____

[问题3] 小张是B公司的开发经理,在开发A市宏观经济数据库原型时,用前端分析工具,按照指标计算模型计算“宏观投资现状”指标,该指标计算模型的输入参数包括如下内容。

- (1) 交易额,信息来源是市国税局和市地税局的税收征管系统。
- (2) 项目立项信息,信息来源是市发改委的立项审批管理系统。
- (3) 资金拨付信息,信息来源是市财政局的统一支付系统。
- (4) 贷款信息,信息来源于银行的信贷管理系统等。
- (5) 资源消耗信息,信息来源于电力公司、自来水公司等。

小张按照输入参数来源单位提供的接口说明取得了输入参数,经过计算得到了“宏观投资现状”指标,但发现该指标与传统方式得到的指标差别很大。

请用200字以内文字指出存在差别的可能技术原因。

答: _____

3.7.4 实验总结

3.7.5 实验评价(教师)

3.8 阅读与分析：为什么自由软件的用户体验差

过去几年中,自由软件的用户体验得到了很大提高,然而整体质量依旧无法与专有软件相比。绝大部分自由软件在规划、设计和编写等各个环节充满各种问题,导致他们的质量不佳,用户体验很差。