第6章
公 共 机 制

本章内容：

- 注解
- 构造型、标记值和约束
- 对注释建模
- 对新的构造块建模
- 对新特性建模
- 对新语义建模
- 扩展UML

通过在整个语言中一致使用的 4个公共机制：详述、修饰、公共划分和扩展机制，可使得UML更加简化。本章说明这些公共机制中的两种机制 —— 修饰和扩展 —— 的用法。【在第2章中讨论这些公共机制。】

注解是一种最重要的能单独存在的修饰。注解是附加在元素或元素集上用来表示约束或注释的图形符号。用注解为模型附加一些诸如需求、观察资料、评论和解释之类的信息。

UML的扩展机制允许以受控的方式对语言进行扩展。这些机制包括构造型、标记值和约束。构造型扩展UML的词汇，允许创建一些新的构造块，这些新构造块是从已有的构造块派生，但针对你的特定问题。标记值扩展 UML构造块的特性，允许在元素的规格说明中创建新的信息。约束扩展UML构造块的语义，允许增加新的规则或修改已存在的规则。使用这些机制对 UML进行裁剪，以满足领域或开发氛围的特殊需要。

6.1 入门

有时必须要在原来的基础上加以润色。例如，在一个工作场地，建筑师可能在建筑物的蓝图上填写一些注解，向建筑工人传达更精致的细节。在录音棚里，作曲家可能从键盘乐演奏者那里得到启发，发明一种能表达一些不寻常效果的新的音乐表示法。在这两种情况下，建筑蓝图和音乐表示法都是已定义好的语言，但有时为了表达一些意图，就必须以一定的受控方式改变或扩展这些语言。

建模完全是为了交流。对于软件密集型系统， UML已经提供了可视化、详述、构造和文档化各种各样的制品所需的各种工具。然而，你可能发现一些要改变或扩展 UML的情况。对人类的语言也始终存在着这样的情况（这也是为什么每年要出版新词典的原因），因为没有哪一种静态语言永远能令人满意地涵盖要交流的每一件事物。当使用像 UML这样的建模语言时，要记住使用这个语言是为了交流，这意味着，除非有被迫偏离的原因，否则要坚持用这种核心语言。当需要在原有的基础上加以润色时，也仅能以受控的方式进行。否则就不可能使每个人都理解

你做的事。

为了有助于说明所建的模型，可以使用 UML提供的注解机制去捕获任意的注释或约束。注解可以描述在软件开发的生命周期内扮演重要角色的制品（如需求），也可以简单地描述自由形态的观察资料、评论和解释。

UML为注释和约束提供了图形表示，称之为注解（ note），如图6-1所示。这种表示法允许直接对注释进行可视化。与适当的工具相结合，注解也可以作为连接或嵌入到其他文档中的占位符。

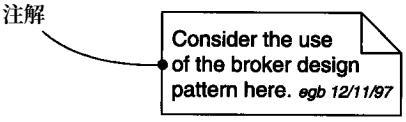

图6-1 注解

构造型、标记值和约束是 UML提供的用以增加新的构造块、创建新的特性以及详述新的语义的机制。例如，如果对网络建模，可能要分别给出路由器和集线器的表示符号；你可以用构造型化节点表示这些的事物，使它们就好像原有的构造块一样。类似地，项目发布组的成员要负责装配、测试和配置发布，可能要跟踪版本号和各个主要子系统的测试结果；对此就可以用标记值把这些信息附加到模型上。最后，如果对要求严格的实时系统建模，可能要用时间预算和最后完成期限来修饰模型；可以使用约束捕获这些计时需求。

UML为构造型、标记值和约束提供了文字表示方法，如图 6-2所示。构造型也允许引入新的图形符号，以便为模型提供可视化提示，以适应特定领域和开发氛围。

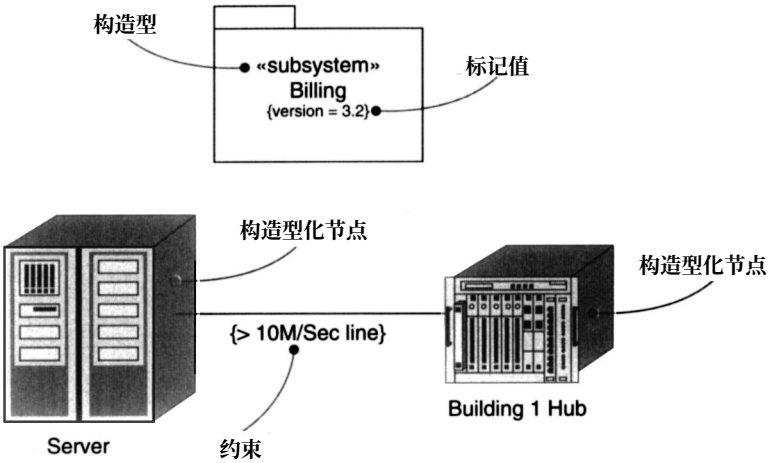

图6-2 构造型、标记值和约束

6.2 术语和概念

注解（*note*）是附加在元素或元素集上用来表示约束或注释的图形符号。在图形上，把注解

画成带有摺角的矩形，在矩形中填写文字或图形注释。

构造型（*stereotype*）是对UML的词汇的扩展，允许创建与已有的构造块相似而针对特定问题的新种类的构造块。在图形上，把构造型表示成用书名号括起来的名称，并把它放在其他的元素名之上。作为一种选择，可以用一种与构造型相联系的新图标表示被构造型化元素。

标记值（*tagged value*）是对UML元素的特性的扩展，允许在元素的规格说明中创建新的信息。在图形上，把标记值表示成用花括号括起来的字符串，并把它放在其他的元素名之下。

约束（*constraint*）是对UML元素的语义的扩展，允许增加新的规则或修改已有的规则。在图形上，把约束表示成用花括号括起来的字符串，并把它放在关联的元素附近，或者通过依赖关系连接到这个（或这些）元素。作为一种选择，可以在注解中表示约束。

1. 注解

表示注释的注解对模型没有什么语义影响，这意味着它的内容不会改变它所依附的模型的含义。这是为什么用注解描述像需求、观察资料、评论和解释之类事物的原因，也是用注解表示约束的原因。

注解可以含有任何的文字或图形。如果实现允许，也可以把 URL（用户需求语言）放到注解中，甚至可以连接或嵌入其他文档。通过利用这种方式，可以用 UML去组织在开发期间生成的或使用的所有的制品，如图 6-3所示。【通过使用在第5章中讨论的依赖，可以把注解依附到多个元素上。】

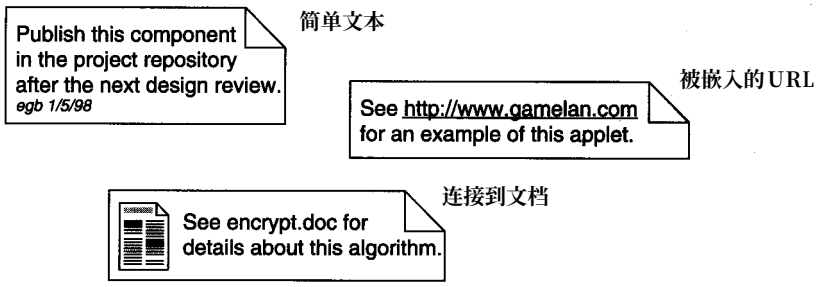

图6-3 注解

注释 UML定义了一个应用到注解上的标准构造型，即需求。这个构造型命名了一个公共的注解种类，即用于陈述某些职责或责任的注解。

2. 其他修饰

修饰是附加到元素的基本表示法上的文字或图形项，用于对元素规格说明的细节进行可视化。例如，关联的基本表示法是一条线，可以用各端的角色或多重性等细节来修饰它。在使用UML时，要遵循如下的一般规则：先对每个元素使用基本表示法，然后仅当有必要表达模型的重要的特殊信息时，才增加其他修饰。【在第5章和第10章中讨论关联的基本表示法和一些修饰。】

大多数修饰是通过在感兴趣的元素附近放一些文字或对基本表示法增加图形符号表示的。然而，有时要用比简单文本或图形符号更能提供细节的事物来修饰元素。对诸如类、构件和节点这样的事物，可以在它们通常的分隔栏的底部增加额外的分隔栏，以填写这种信息，如图 6-4

所示。

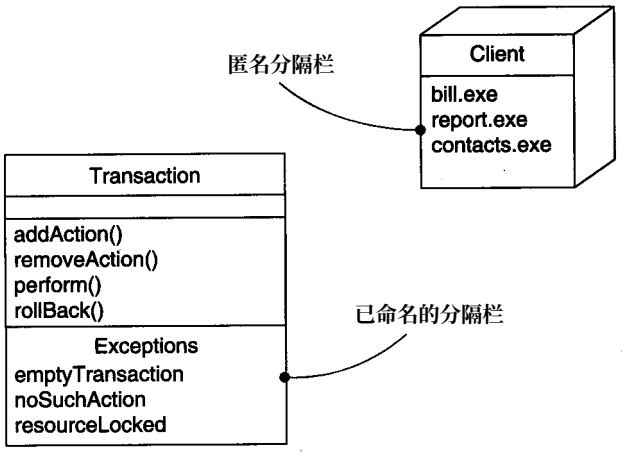

图6-4 额外的分隔栏

注释 除非分隔栏的内容很明显，否则最好对任何额外的分隔栏都显式地命名，以避免含义混淆。此外还建议尽量少使用额外的分隔栏，因为若过度地使用，会造成图形混乱。

3. 构造型

UML为结构性的事物、行为性的事物、成组的事物和注释性的事物提供了一种语言。用这4种基本事物可以表达出绝大多数需要建模的系统。然而，有时要引入能够说清你的领域中的词汇且看起来仍像原有的构造块的新事物。【在第2章中讨论UML中上述的4种基本元素。】

构造型与泛化关系中的父类不一样。确切地讲，可以把构造型看作元类型，因为每一个构造型会创建一个相当于UML元模型中新类的等价物。例如，如果对一个商业过程建模，则将引入像工人、文档和政策这样的事物。类似地，如果正在进行像Rational统一过程这样的开发过程，则将使用边界、控制和实体类来建模。这是构造型的实际价值所在。当对节点或类这样的元素建立构造型时，实际上是通过创建类似于已有的构造块的新构造块来扩展UML，但新构造块有自己的具体特性（各构造型可以提供自己的标记值集）、语义（各构造型可以提供自己的约束）和表示法（各构造型可以提供自己的图标）。【在附录C中总结Rational统一过程。】

最简单的形式是把构造型用一个由书名号括起来的名称表示（如《name》），并且把它放在别的元素的名称之上。可以为构造型定义一个图标，作为可视化提示，并把该图标放在名称的右边（如果用基本表示法来表示元素），或用这个图标作为被构造型化项的基本符号。图6-5说明了这3种方法。【在附录B中讨论UML定义的构造型。】

注解 当为构造型定义图标时，考虑用颜色来作为特征以提供副标题式的可视化提示（但不要过多地使用颜色）。UML允许用任何图形作为图标，如果实现允许，这些图标可以作为简单的工具出现，这样创建UML图的用户就会有描绘事物的“调色板”。对他们来讲，这些被调了色的事物看起来像基本的，但表达的却是具体领域的词汇。【在附录B中讨论UML定义的构造型。】

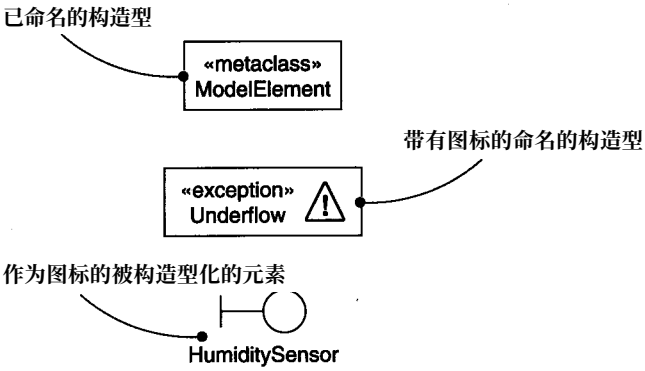

图6-5 构造型

4. 标记值

UML中的每个事物都有它们自己的特性集：类有名称、属性和操作，关联有名称和两个或两个以上的端点（每个端点都有自己的特性）等。用构造型能为 UML增加新的事物；用标记值能为UML的事物增加新的特性。

可以为已存在的UML元素定义标记，还可以定义应用到各构造型的标记，使每一个拥有构造型的事物都有标记值。标记值与类的属性不同。确切地讲，可以把标记值看作是元数据，这是因为它的值应用到元素本身，而不是它的实例。例如，如图 6-6所示，你可能要指定在一个实施图中安装在每一种节点上的处理机数目；如果打算把构件部署在客户机或服务器上，你还可能需要把每个构件构造型化为库。【在第4章和第9章中讨论属性。】

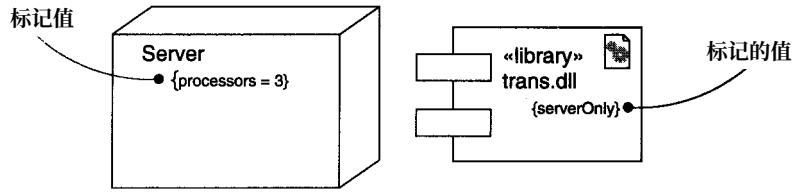

图6-6 标记值

最简单的形式是把标记值表示成一个由括号括起来的串，并放在另一个元素的名称之下。这个串包括一个名称（标记）、一个分隔符（=）和一个（标记的）值。如果一个标记的含义是明确的，就可以指定标记的值。例如，当值是枚举的名称时，就是如此。

注解 标记值的最常用的用途之一是详述与代码生成或配置管理相关的特性。例如，用标记值指定特定类映射到的编程语言。类似地，可以用标记值描述一个构件的作者或版本。【在附录B中讨论UML已定义的标记值。】

5. 约束

UML中的每一个事物都有它自己的语义。泛化意味着运用 Liskov替代原理，连接到一个类的多样的关联表示不同的关系。使用约束，可以增加新的语义或改变已存在的规则。约束详述了一个规范良好的模型必须为真的条件。如图 6-7所示，对于一个给定的关联，可以详述对通信

加密。类似地，可以通过约束说明，在一组关联中同一时刻只有一个关联有效。【当对实时系统建模时，一般用时间和空间约束，这在第23章中讨论。】

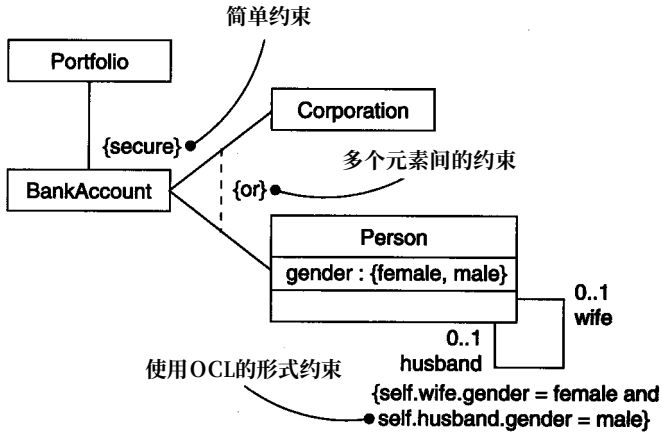

图6-7 约束

注解 可以把约束写成自由形式的文本。若要更精确地详述语义，可以使用UML的对象约束语言（OCL），在《UML参考手册》中对该语言做了进一步的描述。【在附录B中讨论UML已定义的约束。】

一个约束用一个由花括号括起来的串表示，放在相关的元素附近。这种表示法也被用作对元素的基本表示法的修饰，以可视化没有任何图形提示的元素的规格说明部分。例如，用这种表示法来表示关联的一些特性（次序和可变性）。【用依赖可以把约束依附到多个元素上，这在第5章中讨论。】

6. 标准元素

对于类元、构件、关系和一些其他建模元素，UML定义了一些标准构造型。有一个主要为工具建造者感兴趣的标准构造型，使你可对构造型本身建模。【在附录B中总结UML的标准元素，在第9章中讨论类元。】

- 构造型（stereotype）——描述了类元是可以应用到其他元素的构造型。

当要显式地对已经为项目定义的构造型建模时，要使用这个构造型。

UML也描述了一个应用到所有建模元素上的标准的标记值。

- 文档（documentation）——描述了它所依附的元素的注释、描述或解释。

当要把注释直接依附到元素（如类）的规格说明上时，要使用这个标记值。

6.3 普通建模技术

6.3.1 对注释建模

使用注解的最普通的目的是把观察结果、评论或解释以自由的形式写下来。通过把这些注

释直接放在模型中，模型就成了开发过程中创建的各种制品的公共资料库。甚至能用注解把需求可视化，并显式地表示出需求怎样与模型的相关部分相对应。

为了对注释建模，要遵循如下策略：

- 把注释文本放入注解内，并把该注解放于它所对应的元素附近。可以用依赖关系把注解与对应的元素相连接，以更明确地表明关系。
- 要记住，可以按需要隐藏或显示模型中的元素。这意味着不必到处显示依附到可视元素上的注释，而只有在语境中需要交流这种信息时才显露图中的注释。
- 如果注释冗长或者包含比纯文本更复杂的事物，可考虑把注释放在外部的文档中，并把文档连接或嵌入到依附于模型的相应注解中。
- 当建模演化时，保持那些有意义的记录结果，并且这些结果不能从模型本身导出。要舍弃无保留价值的注释。

例如，图6-8显示了一个模型，它是一个正在开发中的类层次，表明了一些形成模型的需求以及一些来自设计观点的注解。【在第5章中讨论简单的泛化，在第10章中讨论泛化的高级形式。】

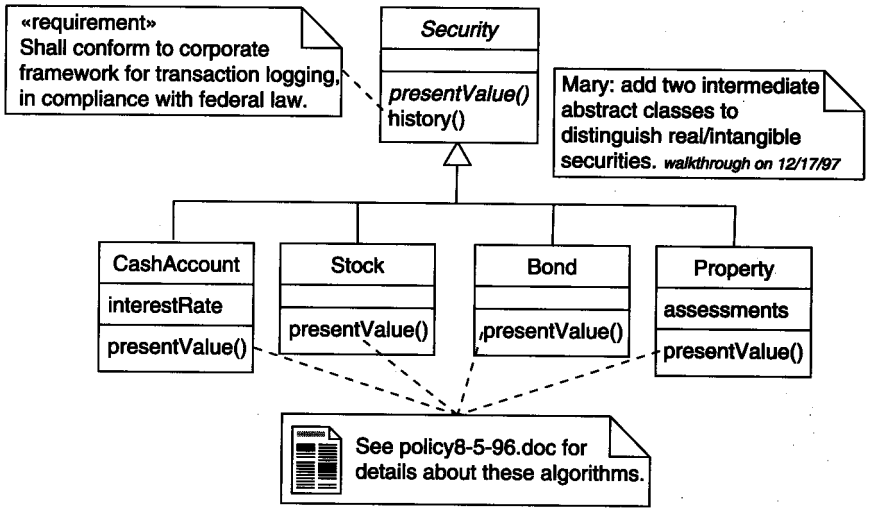

图6-8 对注释建模

在本例中，大多数注释是简单文本（如对 Mary的注解），但其中的一个注释（在图底部的注解）含有一个超链接，用以连接另一个文档。

6.3.2 对新构造块建模

UML的构造块，即类、接口、协作、构件、结点和关联等，一般足以表达要建模的大多数事物。然而，若想扩展建模的词汇，或对经常出现在领域中的一定种类的抽象给出一个清楚的可视化提示，就需要使用构造型。

为了对新构造块建模，要遵循如下策略：

- 要确认用基本的UML已无法表达你要做的事情。如果是常见的建模问题，可能已存在某些

标准的构造型，可满足你的需要。

- 如果确信没有其他的方法能表达这些语义，标识与要建模的事物最相像的 UML中的基本事物（如类、接口、构件、节点、实例连接等），并为该事物定义一个新构造型。要记住，为了具有一般种类的构造型及其他们的特殊构造型，可以定义构造型的层次（但注意层次不要太多）。【在第10章中讨论构造型的构造层次。】
- 通过对构造型定义一组标记值和约束，详述正被构造型化的基本元素本身以外的一般特性和语义。
- 如果想让这些构造型元素有清晰的可视化提示，就要为该构造型定义一个新的图标。

例如，假设现在要用活动图对涉及到教练和运动队在一个体育事件中的工作流的业务过程建模。在这个语境中以可视化的方式区分教练和运动队，并区分他们与领域的其他事物（如事件和部门）是有意义的。如图 6-9所示，有两种突出的事物，即对象 Coach和对象Team的事物。这些不是简单种类的类。准确地讲，现在它们是能在这个语境中使用的基本构造块。可以通过定义教练和运动队构造型，并把它们应用到 UML的类中来创建这些新构造块。在这个图中，被称为 :Coach和 :Team的匿名实例（后者以不同状态显示，这些状态是 unregistered、registered和finished）是通过使用与这些构造型相关的图标来显现的。【在第13章中讨论实例；在第11章中讨论角色；在第19章中讨论活动图。】

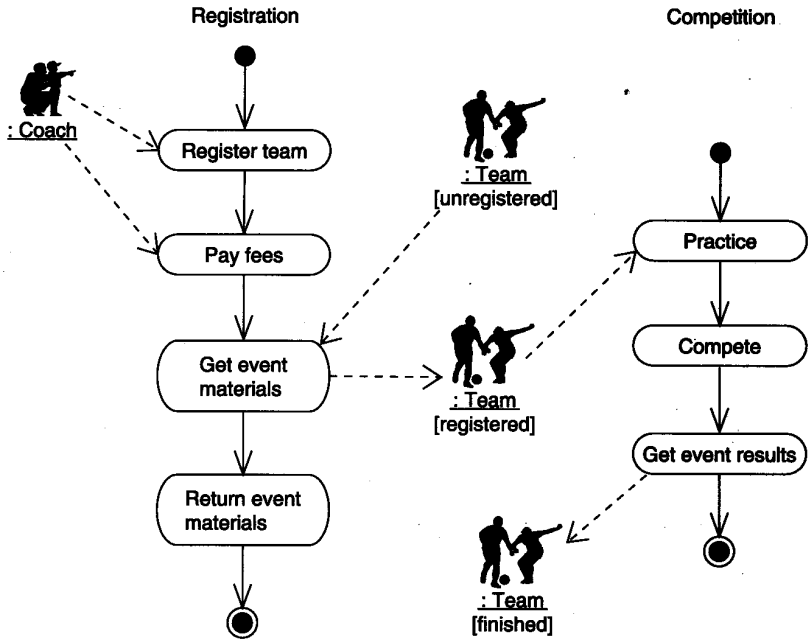

图6-9 对新构造块建模

6.3.3 对新特性建模

UML构造块的基本特性（类的属性和操作以及包的内容等）一般足以表达要建模的大多数

事物。然而，如果要扩展这些基本构造块（或用构造型创建的新构造块）的特性，需要使用标记值。

为了对新特性建模，要遵循如下策略：

- 首先，要确认用基本的UML已无法表达你要做的事情。如果是常见的建模问题，可能已存在一些标准的标记值可满足你的需要。
- 如果确信没有其他的方法能表达这些语义，就对个体元素或构造型增加这种新特性。泛化的应用规则是：为一种元素定义的标记值，可应用到它的子孙。

例如，假设你要用项目配置管理系统来管理你所创建的模型。这意味着在要做的其他事情中还要包括追踪版本号和当前的检入/检出状态，甚至还要包括追踪各子系统的创建或修改日期。虽然可以把这种信息作为标记值，但它不是UML的基本部分，这是因为它是过程所特有的信息。此外，这种信息也不是类的属性。子系统的版本号是它的元数据的一部分，而不是模型的一部分。

如图6-10显示了4个子系统，经扩展后每个子系统都包含有版本号和状态。在子系统Billing中，显示了一个另外的标记值，即当前检出本子系统的人，

注解 可以用工具设置像version和status这样的标记的值。可以把配置管理工具和建模工具结合起来作为开发环境，以此来维护这些值，这样做要胜于用手工设置模型中的这些值。

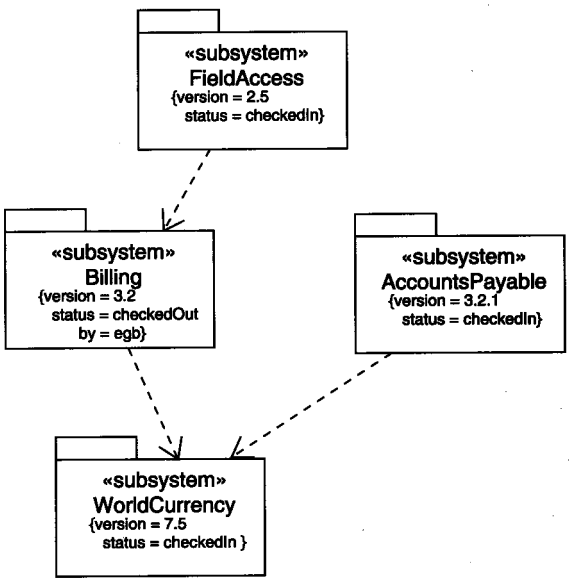

图6-10 对新特性建模

6.3.4 对新语义建模

当用UML创建模型时，要在UML规定的规则下工作。这是件好事情，因为这意味着，可无

歧义地向知道怎样读UML的人交流你的想法。然而，如果你发现自己需要表达 UML中不存在的新的语义，或需要修改UML中的规则，就需要写一个约束。

为了对新语义建模，要遵循如下策略：

- 首先，要确认用基本的UML已无法表达你要做的事情。如果是常见的建模问题，可能存在一些标准的约束，可满足你的需要。
- 如果确信没有其他的方法能够表达这些语义，可以以约束的形式把新的语义写成文本，并把它放在相应的元素附近。通过用依赖关系把约束和相应的元素相连接，能够显示出更明确的关系。
- 如果需要把新语义描述得更精确和更形式化，就用 OCL书写新语义。

例如，图6-11是对一个社团人力资源系统中的一小部分的建模。

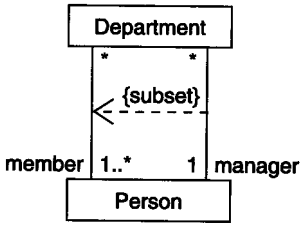

图6-11 对新语义建模

此图表明，每个Person可以是零个或多个Department的成员，每个Department至少有一个Person作为成员。该图还指出了每个Department必须有恰好的一个Person作为主管，每个Person可以是零个或多个Department的主管。所有这些语义都可以用基本的 UML表达。然而，为了断定一个主管也必须是相应的 Department的一个成员，就要涉及到多个关联，这无法用基本的UML表达。为了说明这个不变式，必须写一个约束，以表明主管是 Department的成员的一个子集，用一个从子集到超集的依赖连接这两个关联及该约束。

6.4 提示和技巧

当用注解修饰模型时，要遵循如下策略：

- 仅使用注解来表达那些用现有的UML特征不能简单或有意义地表达的需求、观察资料、评论和解释。
- 把注解作为一种电子粘贴便签，用以跟踪工作的进展。

当绘制注解时，要遵循如下策略：

- 注意不要因使用大块的注释而弄乱模型。如果确实需要长的注释，宁可把注解作为一个占位符，用来连接或嵌入包含全部注释的文档。

当用构造型、标记值或约束扩展UML时，要遵循如下策略：

- 对项目中使用的一小部分构造型、标记值或约束进行标准化，避免让个别的开发人员创建

许多新的扩展。

- 为构造型和标记值选择简短和有意义的名称。
- 在精度要求不高的地方，使用自由形式的文本描述约束。若要求更严密，就用 OCL书写约束表达式。

当绘制构造型、标记值或约束时，要遵循如下策略：

- 要少用图形构造型。你可以用构造型完全改变 UML的基本表示法，但这样做可能会使其他任何人都不理解你的模型。
- 对图形构造型，可以考虑使用简单的颜色或阴影，也可以使用较复杂的图标。简单的表示法一般来说是最好的，即使是最单薄的可视化提示对于理解其含义也大有帮助。