

UML 中衍型的精确定义与分析

蒋严冰, 邵维忠, 张 路, 麻志毅

(北京大学信息科学与技术学院 软件研究所, 北京 100871)

摘 要: UML 规范对于衍型这种扩展机制描述的不够清晰和严格, 常被用户和研究者误用, 并且也无法很好地支持可扩展的建模工具的开发. 本文精确地定义 UML 中的衍型以及衍型之间的关系, 在此基础上定义衍型与元模型之间的转化并提出运用衍型的指导规则, 使得 UML 衍型的使用者能够更深入地理解这种扩展机制, 并为支持衍型的建模工具的开发提供可靠的理论基础.

关键词: UML; 扩展机制; 衍型; 形式化; 语法; 语义

中图分类号: TP311 **文献标识码:** A **文章编号:** 0372-2112 (2003) 12A-2104-05

On the Formal Definition and Analysis of Stereotype in UML

JIANG Yan-bing, SHAO Wei-zhong, ZHANG Lu, MA Zhi-yi

(Software Institute, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871, China)

Abstract: Because the concept of stereotype in UML is not precisely defined, users and researchers often misused this kind of extension mechanism of UML, and modeling tools that support stereotype mechanism can not be well developed by the doctrine in UML. In this paper, we formally define the concept of stereotype in UML, and on this base, research the transformation from stereotype to meta model and the instructive rules of using stereotype. The research in this paper can help the user of stereotype understanding this kind of extension mechanism deeply, and provide reliable theory base for the development of modeling tools that support stereotype mechanism.

Key words: UML; extension mechanism; stereotype; formalization; syntax; semantics

1 引言

统一建模语言 UML 在其产生之后的短短几年内成为被软件领域广泛接受的建模语言. 为了适应对不同领域的建模需求, UML 中定义了三种扩展机制: 标记定义、约束和衍型. 这些扩展机制的引入使得 UML 不再是一个封闭的系统, 建模者可以根据实际的建模需求, 选择适当的扩展机制对 UML 进行扩展, 使之更能精确地刻画特殊的建模领域.

UML 中的衍型是一种重要的扩展机制. 衍型的概念是由 Rebecca Wirfs-Brock 在文献[1]中首先引入的. 她根据对象在系统中完成的责任, 使用衍型将对象分类. 与类的分类方式不同的是, 类的分类方式针对于问题域, 而衍型的分类方式针对解决方案的结构. UML 沿用了衍型的这种直接而明确地向一个概念加入附加的分类信息的方式. 所不同的是, Wirfs-Brock 的衍型是对对象分类, 而 UML 中的衍型是对类分类的机制.

UML 中的衍型是定义了附加的值(基于标记值)、附加的约束以及非强制的新的图形表示的建模元素^[2]. 虽然 UML 规定了衍型的意义与用法, 但规范中的不同部分存在对同一概

念的不同的解释, 很容易给读者一种不精确、模糊甚至是矛盾的印象. 另外由于 UML 中缺少一种对衍型精确的、形式化的描述, 使得 UML 1.X 中的有关衍型的部分晦涩难懂. 并且 UML 只给出了笼统模糊的衍型定义, 没有将衍型细致地分类. 也没有对衍型的运用提出细致的指导规则, 从而使得衍型在实际建模中往往具有很大的不可操作性和不确定性, 常被用户和研究者误用, 并且也无法很好地支持可扩展的建模工具的开发.

自 UML 规范发布后, 许多研究者对衍型扩展机制进行了研究, 他们普遍认为规范中对衍型定义得不够精确并且缺少必要的应用指导, 并提出了各自不同的对衍型的理解, 并试图改进规范中对衍型的描述^[3]. Stefan Berner 等人在文献[4]中将衍型分为四类: 装饰型衍型、描述型衍型、限制型衍型和优化型衍型. 装饰型衍型改变的是语言的具体的语法; 描述型衍型扩展语言的抽象语法以便表达附加的信息; 限制型衍型扩展语言的语法并对这些扩展加入语义; 优化型衍型将修改语言元素的语义. Stefan Berner 等人的工作可以为 UML 规范中衍型的不同定义作出诠释, 同时比较全面地概括了衍型的种类,

收稿日期: 2003-10-09; 修回日期: 2003-12-24

基金项目: 国家自然科学基金(No. 60073015); 国家高技术研究发展计划(863 计划)课题(No. 2001AA 113070); 国家重点基础研究发展规划(973 计划)项目(No. 2002CB312000)

并给出了正确运用每一类衍型的指导规则。

Ansgar Schleicher 等人在文献[5]中在 Stefan Bemer 等人工作的基础上,进一步讨论了对四种衍型的工具支持。他们认为目前除了 Rational Rose 可以支持装饰衍型外,大多数建模工具基本上不支持衍型。

Colin Atkinson 等人在文献[3]中,主张扩大衍型的定义范围。在 Colin Atkinson 提出的衍型的概念中,存在三种类型:第一种即 Rebecca Wirfs-Brock 最初提出的对对象分类的衍型;第二种是 UML 中对类分类的衍型;第三种是他们提出的即对对象分类又对类分类的衍型。Colin Atkinson 对衍型的建议使得衍型的概念与表示更加复杂。

在对衍型学术上争论的同时,衍型在扩展 UML 支持特定领域建模的研究与实践中得到了广泛的应用。然而,许多应用是不符合 UML 对衍型的定义的。例如,文献[6]中提到:文献[7]和[8]中在扩展 UML 以支持对 web 应用建模时,错误地将类型作为基类 class 的衍型;文献[9]错误地将一些类标识为衍型。Martin Gogolla 等人在文献[10]中举例说明了衍型在建模实践中的困难。他们认为有必要进行衍型形式化的定义与分析,并在此基础上研究与元模型类图的转换,以消除研究者和用户对衍型的不同理解,但他们并没有提出自己的解决方案。

近期发布的 UML2.0^[11]对衍型做了重大修订,对衍型阐述在一定程度上比 UML1.X 更加清晰和容易理解。但是 UML2.0 仍然没有提供衍型的分类,没有给出衍型之间关系(叠加与继承)的精确约束条件,也没有涉及衍型与元模型之间的转换与约束条件。

总之,造成衍型在实践中容易不恰当地使用的原因有以下几点:(1)受 Wirfs-Brock 方法的影响;(2)对元模型认识不清;(3)UML 规范本身对衍型定义得不够准确;(4)没有形式化的定义;(5)缺少工具支持。

针对 UML 中衍型的问题,为了使得衍型能够在实践中被恰当地使用,本文将在结合有关文献的基础上,运用简单的形式化手段,精确地定义的衍型以及衍型之间的关系,在此基础上定义衍型与元模型之间的转化,并提出运用衍型的指导规则,使得 UML 的衍型的使用者能够将衍型和元模型扩展机制结合起来,从而更深入、全面、准确地理解这种扩展机制,并为支持衍型的建模工具的开发提供可靠的理论基础。

2 衍型的精确定义与运用

本部分将根据 UML1.4 中对衍型的描述,重点研究衍型的精确定义、运用与向元模型的转化。UML1.4 中规定,可以运用图形或表格两种方式对衍型进行定义。对于用户来说,用这两种方式就可以清晰地描述所要构造的衍型。然而,为了深刻地理解衍型的含义,进而对衍型之间的关系进行精确的分析并研究衍型与元模型之间的转化,有必要对衍型进行精确的定义,并在此基础上进行逻辑推导,从而实现对衍型各方面的深入认识和研究。

2.1 衍型与元模型构造物的精确定义

UML 中的衍型涉及到元模型层(M_2)和模型层(M_1)两个

层次。元模型层侧重衍型的声明,而模型层强调衍型的运用。

定义 1 在元模型的层次上,可以形式化的将衍型记为七元组:

$$\text{Stereotype}\langle S, B, P, T, C, N, M \rangle,$$

其中, S 表示衍型名; B 表示基类,基类用来指明该衍型可以应用的模型元素的名字集,而这些名字都是元模型构造物的名字; P 表示其父衍型;父类用来指明该衍型的父衍型的名字集,衍型的父衍型仍然是衍型; T 是标记定义集, $T = \{ \langle \text{tagname}, \text{tagvaluetype} \rangle \}$; 其中, tagname 是标记的名称, tagvaluetype 是标记的类型; C 是约束集,负责定义衍型的良构规则; N 是衍型的表示法描述,负责定义衍型的具体语法; M 是衍型的语义描述。

定义 2 我们将元模型中的元模型构造物定义为六元组:

$MC\langle MN, F, A, C, N, M \rangle$, 其中 MN 表示元模型构造物名, F 表示该类的父类集, A 表示属性集, C 表示约束集,也即静态语义描述, N 表示元模型构造物的表示法描述, M 表示元模型构造物的语义描述,即这种元模型构造物的含义。

2.2 衍型的运用

在模型层次(M_1)上,衍型用来标记建模元素,以便使它们具有新的元模型虚拟结构的实例的特征^[2]。可以定义运算 ADD, 用来形式化地定义将衍型附加给一个建模元素。模型层的任一建模元素都可以映射为元模型层的元模型构造物,这种映射可以记为 $f_{M_1 \rightarrow M_2}(e_{M_1}) = e_{M_2}$, 其中 e_{M_1} 表示在 M_1 层的建模元素, e_{M_2} 表示在 M_2 层的元模型构造物, $f_{M_1 \rightarrow M_2}$ 表示它们之间的映射。

定义 3 在这里为了说明衍型的运用,可以将模型层模型元素记为一个二元组:

$$ME = \langle N, S \rangle, \text{ 其中 } N \text{ 是模型元素的名称; } S \text{ 是衍型名。}$$

$f_{M_1 \rightarrow M_2}(ME) = MC\langle MN, F, A, C, N, M \rangle$, MC 表示模型元素 ME 的元模型构造物。

$$\text{ADD}(ME\langle N, S \rangle, \text{Stereotype}\langle S_2, B, P, T, C_2, N, M \rangle) = ME'\langle N', S' \rangle;$$

其中, $N' = N$; $S' = S_2$; $f_{M_1 \rightarrow M_2}(ME'\langle N', S' \rangle) = \text{Stereotype}\langle S_2, B, P, T, C_2, N, M \rangle$ 。

在运用衍型的时候,并不是任何衍型都能标记任何的建模元素。衍型所标记的建模元素的元模型构造物应该是衍型的基类或其子类。形式化地记为: $f_{M_1 \rightarrow M_2}(ME\langle N, S \rangle) = B$ 或 $B \triangle f_{M_1 \rightarrow M_2}(ME\langle N, S \rangle)$, 表示 B 是 $f_{M_1 \rightarrow M_2}(ME\langle N, S \rangle)$ 的父类。

3 衍型到元模型扩展机制的转化

为了研究衍型扩展机制和元模型扩展机制之间的关系,同时也为了使得建模工具能够支持衍型,有必要研究衍型到元模型的转化。

Ansgarz 在文献[5]中认为,定义衍型等价于在元模型中定义初始元模型构造物的子类。

转换 1 形式化地,我们给出映射 $f_{S \rightarrow M}(\text{Stereotype}\langle S, B, P, T, C, N, M \rangle) = MC\langle MN, F, A, C', N', M' \rangle$, 将衍型转变

为元模型. 其中: $MN = S; F = B, T \rightarrow A, C \rightarrow C', N \rightarrow N', M \rightarrow M'$. 这种转换的意义是, 当衍型转换为元模型构造物时, 衍型名变为元模型构造物名, 衍型的基类变为元模型构造物的父类, 对于带有父衍型的衍型的转化将在后面讨论. 衍型的标记值将变成元模型构造物的属性, 衍型的约束将变成元模型构造物的约束, 衍型的表示法将变成元模型构造物的表示法, 衍型的语义将变成元模型构造物的语义(如图 1 所示).

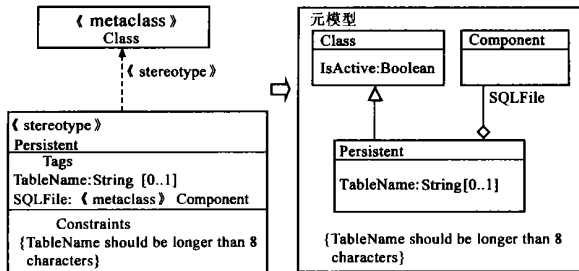


图 1 衍型与元模型扩展机制的转化

由此可见, 任何的衍型都可转化为相应的元模型. 但较之元模型的扩展方法, 衍型只能是对 UML 中已经存在的元模型构造物添加某些属性和约束, 而不能减少某些属性和约束或者是创建新的元模型构造物. 衍型的作用在于使得建模者可以不对元模型做出大量的改进就可以实现对 UML 在语法语义上的扩展, 可以屏蔽许多元模型的细节而构造出许多新的建模元素.

4 衍型的叠加

4.1 衍型叠加的定义

两个以上的衍型可以同时叠加在一个建模元素之中, 称之为衍型的叠加. 衍型的叠加可以使得一个建模元素同时具有两个以上的衍型所描述的特征. 如图 2 所示.

这里仅讨论两个衍型的叠加, 两个以上衍型的叠加可以此类推.

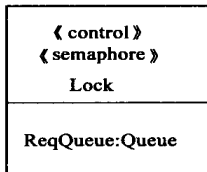


图 2 衍型的叠加

定义 4 我们将衍型叠加的定义如下:

$ADD(ME \langle N, S \rangle, Stereotype1 \langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, Stereotype2 \langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle) = ME' \langle N', S' \rangle$; 表示将两个衍型 Stereotype1 和 Stereotype2 应用到建模元素 ME 之后, 原来的建模元素 ME 就变成了 ME'.

其中, $N' = N$, 表示叠加后的建模元素同原来的建模元素的名字相同; $S' = \{S_1, S_2\}$, 表示叠加后的建模元素的衍型名是两个衍型名的集合.

若将标记了这两个衍型的模型元素映射到元模型层, 就得到这两个衍型组成的集合, 形式化地记为: $f_{M_1 \rightarrow M_2}(ME' \langle N', S' \rangle) = \{Stereotype1 \langle S_1, B_1, P_1, T_1, C_2, N_1, M_1 \rangle, Stereotype2 \langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\}$.

上面的式子的意思是, 当一个模型元素标记了两个衍型后, 它就不再是原来元模型构造物的实例, 而是标记它的两个衍型的共同的实例.

4.2 衍型叠加的约束条件

并不是任何两个衍型都可以叠加在任何一个模型元素之上的, 衍型的叠加需要一定的约束条件.

首先, 每一个衍型所标记的建模元素的元模型构造物应该是该衍型的基类或是其基类的子类. 形式化地记为: $f_{M_1 \rightarrow M_2}(ME \langle N, S \rangle) = B_1$ 或 $B_1 \triangle f_{M_1 \rightarrow M_2}(ME \langle N, S \rangle)$, 并且 $f_{M_1 \rightarrow M_2}(ME \langle N, S \rangle) = B_2$ 或 $B_2 \triangle f_{M_1 \rightarrow M_2}(ME \langle N, S \rangle)$.

其次, 模型元素所叠加的衍型的各个方面不能冲突. 所允许叠加的衍型与类型没有关系, 但必须不能发生冲突.

4.3 衍型叠加到元模型的转换

转换 2 可以定义映射 $f_{S \rightarrow M}$ 将元模型层次的衍型集转化成为一个元模型构造物: $f_{S \rightarrow M}(\{Stereotype1 \langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, Stereotype2 \langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\}) = MC \langle MN, F, A, C, N, M \rangle$

其中: $MN = S_1 + S_2$ 表示可以用两个衍型名字的组合命名新的元模型构造物; $F = \{B_1, B_2\}$ 表示元模型构造物的父类是两个衍型的基类; $T_1 \cup T_2 \rightarrow A$ 表示两个衍型的标记定义的并集转变为元模型构造物的属性集. 如果两个衍型中存在名称和类型一样的标记定义, 那么在元模型构造物中只保留一个; 如果两个衍型中存在名称一样的标记定义, 但类型不同, 那么在元模型构造物中两个都保留; $C_1 \cup C_2 \rightarrow C$ 表示两个衍型的约束集的并集转变为元模型构造物的约束集; $N_1 \cup N_2 \rightarrow N$ 表示两个衍型的表示描述的并集转换为元模型构造物的表示描述; $M_1 \cup M_2 \rightarrow M$ 表示两个衍型的语义描述的并集转换为元模型构造物的语义描述.

图 3 是衍型叠加转换到元模型叠加的示意图. 图中的左侧是模型层中叠加了两个衍型的模型元素. 右侧是与其对应的元模型的衍型和元模型构造物. 右上角为模型元素所叠加的两个衍型 a、b 与其基类 B_1, B_2 , 可以认为标记了这两个衍型的模型元素是这两个元模型层的衍型叠加的实例. 可以通过转换 $f_{S \rightarrow M}$ 将衍型 a 和 b 的叠加转变为元模型构造物 ab, 模型层叠加了衍型 a 和 b 的模型元素也是元模型构造物 ab 的实例.

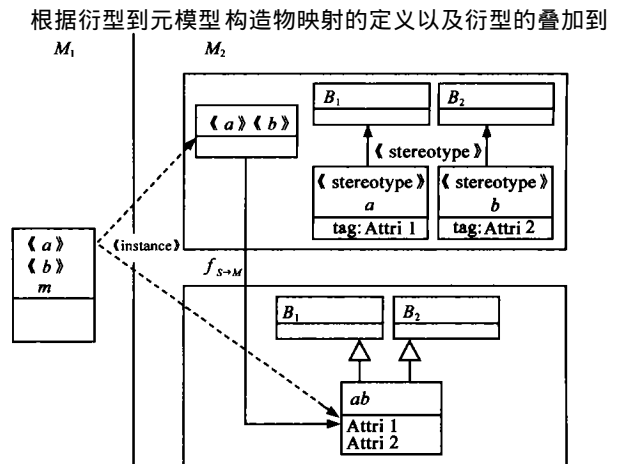


图 3 衍型叠加的转换

元模型构造物映射的定义可以推出这样的推论:

推论 1 $\{f_{S \rightarrow M}(\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle), f_{S \rightarrow M}(\text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle)\} \Delta f_{S \rightarrow M}(\{\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\})$, 其意义是两个衍型叠加之后转化成的元模型构造物是多重泛化了两个衍型分别转化的元模型构造物。

证明 根据衍型转化为元模型构造物的定义, $f_{S \rightarrow M}(\text{Stereotype}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle) = MC\langle MN_1, F_1, A_1, C_1, N_1, M_1 \rangle$, 其中: $MN_1 = S_1; F_1 = B_1, T_1 \rightarrow A_1, C_1 \rightarrow C_1, N_1 \rightarrow N_1, M_1 \rightarrow M_1$; $f_{S \rightarrow M}(\text{Stereotype}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle) = MC\langle MN_2, F_2, A_2, C_2, N_2, M_2 \rangle$, 其中: $MN_2 = S_2; F_2 = B_2, T_2 \rightarrow A_2, C_2 \rightarrow C_2, N_2 \rightarrow N_2, M_2 \rightarrow M_2$.

根据衍型叠加转换到元模型的定义, $f_{S \rightarrow M}(\{\text{Stereotype}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\}) = MC\langle MN, F, A, C, N, M \rangle$.

其中: $MN = S_1 + S_2, F = \{B_1, B_2\}, T_1 \cup T_2 \rightarrow A, C_1 \cup C_2 \rightarrow C, N_1 \cup N_2 \rightarrow N, M_1 \cup M_2 \rightarrow M$.

由此可以看出, 元模型构造物 MN 具有元模型构造物 MN_1 和 MN_2 共同的特征, 因此元模型构造物 MN 多重继承了元模型构造物 MN_1 和 MN_2 , 证毕。

推论 2 衍型的叠加符合交换律: $\text{ADD}(ME, \text{Stereotype1}, \text{Stereotype2}) = \text{ADD}(ME, \text{Stereotype2}, \text{Stereotype1})$. 证明略。

5 衍型的泛化

在衍型的声明的时候, 某些情况下为了复用已经存在的衍型定义, 可以用新的衍型泛化已经存在的衍型, 如图 4 所示. 在模型层运用衍型的时候也存在衍型的泛化问题. 两个标记了同种衍型的模型元素之间可能存在泛化关系, 这取决于衍型的基类或衍型的定义. 这里我们讨论的是衍型定义时的泛化问题。

首先给出衍型泛化的定义:

定义 5 $\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle \Delta \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle$, 则必有 $B_1 \Delta = B_2, P_2 = \text{Stereotype1}, T_1 \subseteq T_2, C_1 \subseteq C_2, N_1 \subseteq N_2, M_1 \subseteq M_2$.

该公式的意义是, 如果衍型 Stereotype2 泛化了衍型 Stereotype1, 那么必有 Stereotype2 的基类 B_2 泛化或等于 Stereotype1 的基类 B_1 , 并且 B_2 的父衍型设置为 Stereotype1, Stereotype2 的标记定义集 T_2 包含了 Stereotype1 的标记定义集 T_1 , Stereotype2 的约束集 C_2 包含了 Stereotype1 的约束集 C_1 , Stereotype2 的表示描述 N_2 包含了 Stereotype1 的表示描述 N_1 , Stereotype2 的语义描述 M_2 包含了 Stereotype1 的语义描述 M_1 .

衍型之间存在多重泛化, 下面是多重泛化的定义:

定义 6 若 $\{\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\} \Delta \text{SubStereotype}\langle S, B, P, T, C, N, M \rangle$, 则必有 $B_1 \Delta = B, B_2 \Delta = B, P = \{\text{Stereotype1},$

$\text{Stereotype2}\rangle, T_1 \subseteq T, T_2 \subseteq T, C_1 \subseteq C, C_2 \subseteq C, N_1 \subseteq N, N_2 \subseteq N, M_1 \subseteq M, M_2 \subseteq M$.

该公式的意义是, 如果衍型 SubStereotype 多重泛化了衍型 Stereotype1 和 Stereotype2, 那么 Stereotype 的基类 B 泛化或等于 Stereotype1 的基类 B_1 以及 Stereotype2 的基类 B_2 , Stereotype 的父衍型为 Stereotype1 和 Stereotype2, Stereotype 的标记集 T 包含 Stereotype1 的标记集 T_1 和 Stereotype2 的标记集 T_2 , Stereotype 的约束集 C 包含 Stereotype1 的约束集 C_1 和 Stereotype2 的约束集 C_2 , Stereotype 的表示描述 N 包含 Stereotype1 的表示描述 N_1 和 Stereotype2 的表示描述 N_2 , Stereotype 的语义描述 M 包含 Stereotype1 的语义描述 M_1 和 Stereotype2 的语义描述 M_2 .

推论 3 $\{\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\} \Delta$

$\text{SubStereotype}\langle S, B, P, T, C, N, M \rangle = (\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle \Delta \text{SubStereotype}\langle S, B, P, T, C, N, M \rangle) \cap (\text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle \Delta \text{SubStereotype}\langle S, B, P, T, C, N, M \rangle)$. 这一公式的意义是, 若衍型 SubStereotype 多重泛化了衍型 Stereotype1 和 Stereotype2, 那么衍型 SubStereotype 泛化了衍型 Stereotype1, 并且衍型 SubStereotype 泛化衍型 Stereotype2. 证明略。

推论 4 若 $\{\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle, \text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle\} \Delta \text{SubStereotype}\langle S, B, P, T, C, N, M \rangle$, 那么 $f_{S \rightarrow M}(\text{subStereotype}\langle S, B, P, T, C, N, M \rangle) = MC$, 且 $\{f_{S \rightarrow M}(\text{Stereotype1}\langle S_1, B_1, P_1, T_1, C_1, N_1, M_1 \rangle), f_{S \rightarrow M}(\text{Stereotype2}\langle S_2, B_2, P_2, T_2, C_2, N_2, M_2 \rangle)\} \Delta MC$. 其意义是, 如果衍型 SubStereotype 多重泛化了衍型 Stereotype1 和 Stereotype2, 衍型 SubStereotype 可转变为元模型构造物 MC , 那么元模型构造物 MC 多重泛化了由衍型 Stereotype1 和 Stereotype2 转变的元模型构造物。

如图 5 所示, 衍型 a, b 的基类分别是 B_1 和 B_2 , 衍型 c 多重泛化了衍型 a 和 b . 衍型 a 可以转化为元模型构造物 a , 衍型 b 可以转化为元模型构造物 b , 衍型 c 可以转化为元模型构造物 ab , 那么元模型构造物 ab 多重泛化了元模型构造物 a 和 b . 证明略。

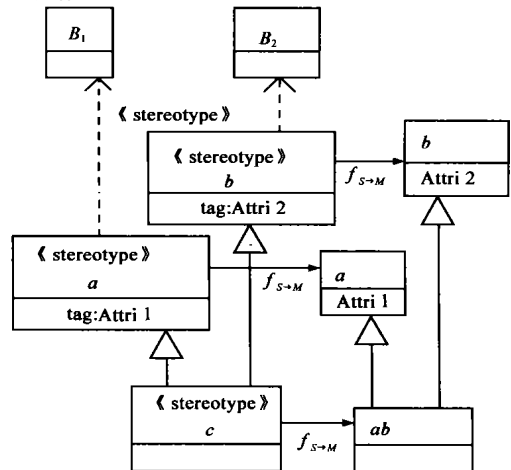


图 5 衍型的多重泛化到元模型构造物的转化

6 应用

可以将以上对衍型的精确定义与分析应用到支持衍型的可扩展工具的开发中。如图 6 所示, 用户根据衍型编辑器及衍型泛化编辑器, 编制出适合特定领域的衍型, 然后带入衍型检查器检查检查是否正确, 这时会用到 2.1 和 5 节中的约束规则。经过衍型检查器检查正确后, 存储在衍型库中。衍型到元模型转换器将衍型库中的衍型转化等价的元模型(这时会用到 3.4.3 和 5 节中的衍型到元模型的转换规则), 并存储在元模型库中。用户在运用自定义的衍型进行建模时, 首先在衍型应用器中用衍型库中的衍型标记模型元素(包括衍型的叠加), 再由应用检查器检查这次标记是否符合规则(这时会用到 2.2 和 4.2 节中的衍型运用约束)。对于符合规则的衍型将通过实例化器实例化为模型片段, 并存储起来。

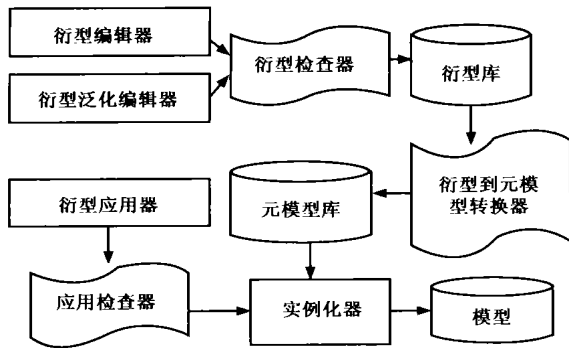


图 6 支持衍型的可扩展工具原理图

7 结束语

针对当前版本的 UML 的衍型的定义过于笼统、容易用错以及缺少可操作性等问题, 本文首先运用浅显的形式化的手段, 给出了衍型形式化定义, 并在此基础上讨论了衍型到元模型扩展机制的转换, 并进一步研究了衍型的叠加与泛化关系的约束条件以及到元模型的转化等问题。所有这些研究增强了 UML 衍型的可理解性和可操作型, 便于建模者深入理解衍型并正确使用它对 UML 进行扩展, 并且为支持衍型的建模工具的开发奠定了理论基础。

参考文献:

- [1] Wirfs-Brock R, Wilkerson B, Wiener L. Responsibility-driven design: Adding to your conceptual toolkit[J]. ROAD 1, 1994, 2: 27-34.
- [2] OMG Unified Modeling Language Specification 1.3, 1.4[EB/OL]. <http://www.omg.org, 1999-06-08/2001-09-06>
- [3] Colin Atkinson, Thomas Kvhue, Brian Henderson-Sellers. Stereotypical

encounters of the third kind[A]. UML 2002-The Unified Modeling Language, 5th International Conference Proceedings[C]. Germany: Springer-Verlag, 2002. 100-114.

- [4] Stefan Bemer, Martin Glinzand, Stefan Joos. A classification of stereotypes for object-oriented modeling languages[A]. UML'99-The Unified Modeling Language: Beyond the Standard[C]. Germany: Springer-Verlag, 1999. 249-264.
- [5] Ansgar Schleicher, Bernhard Westfechtel. Beyond stereotyping: Meta-modeling approaches for the UML[A]. 34th Annual Hawaii International Conference on System Sciences (HICSS-34)[C]. USA: IEEE, 2002. 3051.
- [6] DSTC Pty Ltd. UML 2.0 Infrastructure Revised Submission[EB/OL]. <http://www.community-ml.org/DSTC.htm>, 2002-09-06
- [7] Baresi L, Garzotto F, Paolini P. Extending UML for modeling web applications[A]. 34th Annual Hawaii International Conference on System Sciences (HICSS-34)[C]. USA: IEEE, 2002. 3055.
- [8] Connallen J. Building Web Applications with UML[M]. USA: Addison-Wesley, 2000.
- [9] Sauer S, Engels G. UML-based behavior specification of interactive multimedia applications[A]. IEEE Symposium on Visual Languages 2001[C]. USA: IEEE, 2001. 248-255.
- [10] Martin Gogolla, Brian Henderson-Sellers. Analysis of UML stereotypes within UML metamodel[A]. UML 2002-The Unified Modeling Language, 5th International Conference Proceedings[C]. Germany: Springer-Verlag, 2002. 84-99.
- [11] OMG UML 2.0 Infrastructure, 3rd Revision[EB/OL]. <http://www.omg.org, 2003-03-01>.

作者简介:



蒋严冰 男, 1975 年 1 月生于山东省泰安市, 现为北京大学信息科学技术学院博士研究生, 研究方向: 面向对象的方法与技术、建模语言、建模工具等。

邵维忠 男, 1946 年 12 月生于山东省平度市, 教授, 博士生导师, 主要研究方向: 软件工程环境、面向对象的方法与技术、建模语言与工具、人工智能等。

张路 男, 1973 年 1 月生于四川省绵阳市, 博士, 讲师, 研究方向包括: 软件复用、程序理解、建模工具、软件工程环境等。

麻志毅 男, 1963 年 4 月生于内蒙古自治区赤峰市, 博士, 副教授, 主要研究方向: 软件工程、面向对象的方法与技术、计算语言学等。