

MDA 模型转换的同步技术*

杨磊,余金山

(华侨大学 计算机学院,福建 泉州 362011)

摘要: 给出一种源模型和目标模型间的链接技术及其实现方法,该方法能很好地实现源模型和目标模型间的双向可追踪。既可以对源模型的更新执行正向转换,还可以实现目标模型向源模型的逆向转换。

关键词: 模型驱动;模型转换;同步;可追踪性

中图分类号: TP311.5

文献标识码: A

文章编号: 1674-7720(2011)09-0077-02

A synchronization technique for model transformation in MDA

Yang Lei, Yu Jinshan

(College of Computer Science and Technology, Huaqiao University, Quanzhou 362011, China)

Abstract: This paper presents a link technique of source model and target model and its implementation. The method can well realize two-way traceability between the source and the target. It can not only synchronize the target when any modification is made to source model, but also can achieve the reverse conversion from target to source.

Key words: MDA; model transformation; model synchronization; tracibility

模型驱动架构(MDA)是继 CORBA、UML 之后由 OMG 推出的重要的软件开发方法学^[1-3]。MDA 对软件开发、异构平台的集成、互操作以及对不断涌现出的软件新技术和新平台的自治性等方面都有着巨大的影响^[1-2]。

MDA 的核心问题是模型转换。其基本方法是:先抽象出与实现技术无关,完整描述业务的平台无关模型(PIM),并针对不同的实现制定多个映射规则,再将平台无关模型转换成与具体实现技术相关的应用模型(PSM)。由于建模过程和用户业务逻辑固有的变动性,在模型转换过程中必须保持源模型和目标模型的同步,这对 MDA 的模型转换的同步性研究具有重要意义。

本文提出一种创建中间转换对象的方法,将模型转换信息保存在中间对象中,使模型转换具有正、反向可追溯性,以较好实现源模型和目标模型转换的同步。并从通用性和模型转换的同步性的角度对 MOF 元模型和 OCL 进行了改进。

1 模型转换同步性的中间转换对象技术

模型转换过程中,由于建模过程和用户业务逻辑固有的变动性,一项重要的实际工作是必须实现源模型和

目标模型之间的正向和反向的完全同步。通常的做法是为源模型和目标模型中相应元素之间的所有关系建立元素链接,并把链接信息添加到源模型或目标模型中。这样做不但破坏了模型的纯洁性,而且当源模型和目标模型之间不是一对一时容易引起混乱。

方法(1)是在源模型中创建一系列标记(tag),这些标记记录了转换的信息。图 1 所示的转换必须建立表 1 所示的两个标记。

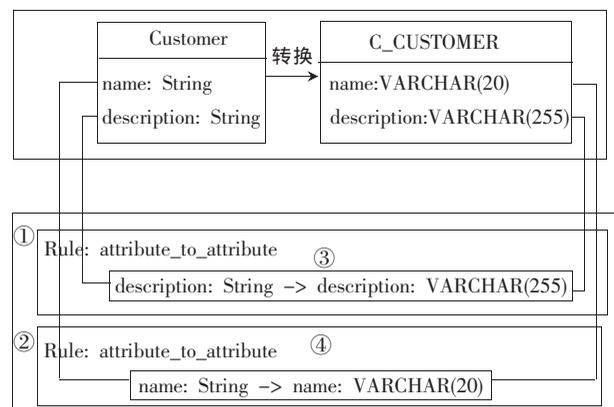


图 1 用中间对象保存模型转换信息(①、②为转换实例,③、④为转换对象信息)

* 基金项目:福建省自然科学基金(A080013)

技术与方法 Technique and Method

表 1 模型转换的信息标记

标记	源	目标	值
tag1	name:String	name:VARCHAR	20
tag2	description:String	description:VARCHAR	255

这些标记是转换信息,并不属于源模型。把它们加入到源模型中不但破坏了源模型的纯洁性,而且当一个PIM要转换成几个PSM时,大量的标记将给PIM带来很多额外的负担。

方法(2)把标记信息保存在目标模型中。NetBeans开发平台就使用这种方法。但该方法的缺点与方法(1)类似,而且在执行第一次转换之前,因为目标模型还没有生成,所以只有在执行过一次系统的默认转换之后,才能在生成的目标模型中保存标记信息。

而较好的解决方法是:(1)让每条被执行的转换规则创建一个转换实例,每个转换实例包含一个或多个转换对象;(2)在每个转换对象中存储一条转换信息,即源模型元素和目标模型元素的链接信息;(3)建立一个中间对象模型,把转换实例及转换对象保存在这个模型中。如图1所示,设要把源模型Customer转换到SQL模型C_CUSTOMER,attribute_to_attribute是负责把源模型中的String类型属性转换到SQL模型的VARCHAR类型属性的转换规则。则当对Customer的描述属性应用规则attribute_to_attribute时,生成转换实例①,实例①包含一个转换对象信息③,在信息③中保存转换信息“Customer的String类型的description转换成C_CUSTOMER的VARCHAR(255)类型的description”。同理,对Customer的name属性应用这条转换规则时,生成转换实例②,②包含一个转换对象信息④,而转换对象信息④中保存了转换信息“Customer的String类型的name转换成C_CUSTOMER的VARCHAR(255)类型的name”。并把转换实例及转换对象保存在另外的既不属于源,也不属于目标的模型中。这样不仅能保留完整清晰的转换信息又不破坏源模型和目标模型的纯洁性。

为了实现以上方法,需要对MOF进行扩展,建立一个双向图结构的元模型。如图2所示,ModelElement是一个最通用的模型元素;一个Link引用两个ModelElement,分别为源对象和目标对象;一个Step包含一个或多个Link,则一个Step就可以保存一次模型转换的追溯信息。如果考虑到支持一个链式的转换,那么整个的转换过程相当于有多个Step,因此,为了支持链式转换,用一个包含了Step的有序集合的Trace来实现。

在MOF元模型层次中增加两个新的元素TransformationTraceClass和Relation,并定义两个操作setRelation()和getRelation(),用于读写Relation的值,如图3所示。

在源模型第一次发生模型转换并需要保存相关信息时,调用TransformationTraceClass,并把本次转换的相关

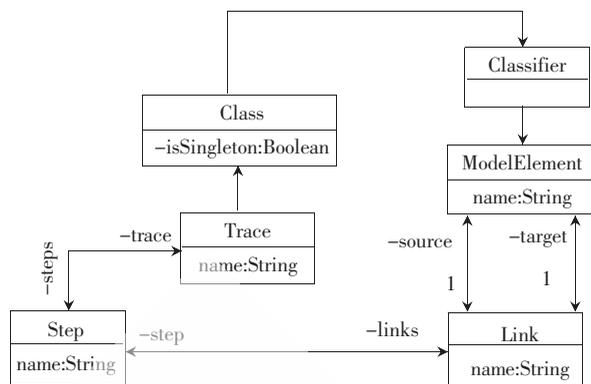


图2 转换信息元模型图



图3 TransformationTraceClass示意图

信息用setRelation()操作写入到Relation的一个元组,此后每进行一个模型元素的转换,就首先到Relation中查找有没有相同模型转换的元组。若有,则把目标元素追加到该元组中;若没有,则在Relation中再创建一个新的元组。即可实现对转换的完全同步支持,同时可将转换关系的信息和模型本身分隔开,实现了关注点的分离,保证了PIM和PSM的清晰性和单纯性。

2 支持模型转换同步的OCL扩展

在MDA中,用UML结合OCL已被认为是开发符合MDA规范的PIM的最佳方案^[1,3]。研究指出:OCL对于模型转换可以发挥重要的作用^[3-4]。但是,如何把OCL应用于模型转换的实际工作却做得较少。由于模型转换的关系信息的复杂性,所以难以用现有OCL来表示。对于用中间转换对象技术保留模型转换的关系信息的OCL实现,对OCL进行如下的扩展:

将图3看成是TransformationTraceClass的图语法。TransformationTraceClass是以由Tuple组成的并以Relation作为其组成元素的。作为一个类,其创建操作必须由创建原语实现,具体格式为:

```
createElement (TransformationTraceClassName : TransformationTraceClass)
```

创建原语实际生成的是对TransformationTraceClass的引用,每个模型转换中的源模型对应生成一个TransformationTraceClass,它在第一次进行模型元素转换

时被创建,且调用 `setRelation()` 操作保存转换信息,调用 `getRelation()` 操作查询转换信息。下面是 `setRelation()` 和 `getRelation()` 的语法。

```
setRelation(newValue, Relation.Tuple.sourceName)
```

`setRelation` 操作类似普通类中对于属性的 `set` 操作,但该原语的操作对象是 `Relation` 中的元组。其含义是:把转换关系信息 `newValue` 写入到以 `sourceName` 为 source 的元组中。

```
getRelation(Relation.Tuple.sourceName)
```

`getRelation` 操作类似普通类中对于属性的 `get` 操作,但其操作对象和返回值均是 `Relation` 中的元组。由于元组只是将一组值集合在一起的一种途径,所以具体使用时元组必须被赋给一个变量。

此外,在模型转换中,对任何模型元素的操作都定义其必须查询相应的 `TransformationTraceClass` 来确定与被修改元素的相关转换信息,从而实现源模型和目标模型的同步。

MDA 堪称是一种革命性的新方法。MDA 的核心问题是模型转换。本文给出一种能对模型转换的向前和向后可追溯性的方法,较好地实现了源模型和目标模型转换的同步。由于 OCL 具有形式化且简洁易懂等众多显

著特征,因而用它来描述 MDA 中的模型具有突出的优点。但 OCL 作为模型转换语言必须进行一定的扩展^[3-4]。本文所做的保存模型同步信息的扩展仅是其中之一。

参考文献

- [1] SENDALL S, KOZACZYNSKI W. Model transformation: The heart and soul of model-driven software development [J]. IEEE Software, 2003(9):42-45.
- [2] 王学斌,王怀民,等.一种模型转换的编织框架[J].软件学报,2006,17(6):1423-1435.
- [3] 陈婧,赵建华,张康康.处理动态行为描述的 MDA 模型转换技术[J].计算机应用与软件,2010,27(4):162-166.
- [4] CARIOU E, MARVIE R, SEINTURIER L, et al. OCL for the specification of model transformation contracts In: Octavian Patrascoiu [R]. OCL and model driven engineering UML 2004 workshop. Lisbon, Portugal, 2004.

(收稿日期:2010-11-08)

作者简介:

余金山,男,1952年生,教授,主要研究方向:软件工程、网络计算、人工智能应用。

杨磊,男,1985年生,硕士研究生,主要研究方向:软件工程、管理信息系统。