

文章编号: 1673- 9469(2010) 04- 0082- 04

## 基于 ATLAS 转换语言的模型转换初探

魏菲<sup>1</sup>, 李慧<sup>2</sup>, 陈月娟<sup>1</sup>, 张京军<sup>1</sup>

(1. 河北工程大学 信息与电气工程学院, 河北 邯郸 056038; 2. 邯郸学院 计算机系, 河北 邯郸 056005)

**摘要:** 提出一种利用 ATLAS 转换语言接收转换信号的模型转换框架, 当平台无关模型 (Platform Independent Model, PIM) 向平台相关模型 (Platform Specialize Model, PSM) 转换时, 利用 ATLAS 转换语言对发出的信号进行模型转换判断, 属于模型转换信号则执行信号任务, 否则进行可行性判断, 若为可行信号, 则返回入口处等待接收新信号, 若为不可行信号, 则向数据库报告并将不可行信号写入日志。

**关键词:** 模型驱动架构; ATLAS 转换语言; 模型转换

**中图分类号:** TP311. 52

**文献标识码:** A

## Model transformation based on ATLAS transformation language

WEI Fei<sup>1</sup>, LI Hui<sup>2</sup>, CHEN Yue-juan<sup>1</sup>, ZHANG Jing-jun<sup>1</sup>

(1. College of info., & Electric Engineering, Hebei University of Engineering, Hebei Handan 056038, China; 2. Department of Computer Science, Handan College, Hebei Handan 056005, China)

**Abstract** A model transformation framework is presented which accept model transformation signals with ATLAS transformation language. When Platform Independent Model (PIM) transform to Platform Specialize Model (PSM), the signal sent by PIM is judged by use of the ATLAS transformation language: if the signal belongs to the model transformation, the signal's task would be executed; otherwise the feasible judgment would carry on. If the signal is viable, it will return to the entrance waiting for the new signal, if not, the signal will be reported to the database and written in log.

**Key words:** MDA; ATL; model transformation

随着软件开发技术的发展, .NET、J2EE、CORBA 等新技术平台不断涌现出来, 然而由于彼此之间无法相互集成和相互操作, 导致了软件开发成本的提高及应用的复杂性增大。国际对象管理组织<sup>[1]</sup> (Object Management Group, OMG) 对此提出了 CORBA 互操作标准, 但是仅仅依靠孤立的接口标准并不能满足跨平台互操作的需求<sup>[1-2]</sup>; 随后 OMG 又于 2001 年 7 月发布了模型驱动架构 (Model Driven Architecture, MDA), 研究重点转为模型的建立及转换<sup>[3]</sup>。目前关于模型建立的研究已日益成熟, 但模型转换的研究尚处于探索阶段<sup>[4-6]</sup>。本文利用 ATLAS 转换语言 (ATLAS Transformation Language, ATL) 提出了一个接收模型转换信号进行模型转换的流程框架, 为今后的模型转换开发提供

借鉴与参考。

### 1 模型驱动架构简介

MDA 是近年来软件工程中研究和实践的热点, 模型在软件开发过程中占有重要的地位。MDA 开发过程的核心思想是: 对系统进行需求分析和设计, 建立反映系统核心业务思想的平台无关模型 PIM (Platform Independent Model), 然后 PIM 经过 MDA 工具的转换生成基于特定平台的平台相关模型 PSM (Platform Specialize Model), PSM 再经过 MDA 工具的转换生成源代码, 最后进入代码测试和维护, 如图 1 所示。

收稿日期: 2010- 06- 20

基金项目: 河北省自然科学基金项目 (F2009000852); 河北省教育厅自然科学研究指导项目 (Z2009107)

特约专稿

作者简介: 魏菲 (1985- ), 女, 河北武安人, 硕士研究生, 从事软件工程方面的研究。

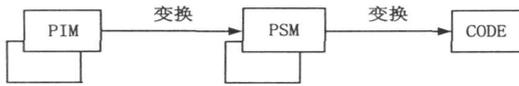


图1 MDA主要开发过程

Fig.1 The MDA of main development process

## 1.1 模型驱动架构的主要模型

1) 平台无关模型 PIM: 平台无关模型 PIM 是反映系统的主要业务的模型, 是对业务需求的一种纯理论的反映, 根据准确的需求分析得到的结果来进行建模的, 这个模型设计好后, 可以供其它系统重复使用。

2) 平台相关模型 PSM: 平台相关模型 PSM 是将 PIM 通过模型规则转换成与特定技术和平台相关的模型, 比如基于 .NET 平台、Spring 平台等。PIM 可以转换成一个或多个 PSM。

## 1.2 模型驱动架构的核心技术<sup>[2]</sup>

建模和模型映射技术是 MDA 的核心技术, UML(Unified Modeling Language), MOF(Meta Object Facility) 及 CWM(Common Warehouse Metamodel) 是 MDA 主要使用的三项建模标准, 作为 PIM 及 PSM 的建模基础。

UML—统一建模语言: UML 是一种元模型, UML 模型的表达能力很强, 会避免建模过程中不必要的信息丢失。

MOF—元对象设施: MOF 用来定义面向对象元模型的基本元素、语法和结构<sup>[4]</sup>。它是一种面向对象的元—元模型, 为规约面向对象元模型定义了一种公共的抽象语言。

CWM—公共仓库元模型: CWM 是一组元模型, 目的是为了在数据仓库工具、数据仓库平台和数据仓库存储之间建立一个商务智能元数据的交换机制<sup>[5]</sup>。它覆盖了数据仓库应用的整个生命周期。

## 2 ATLAS 转换语言

ATL 着重于模型与模型之间的转换, 其语法结构是基于 OCL 2.0 定义的, 适用于表达 MDA 框架中的模型转换。ATL 主要分为三种类型: ATL 模块, ATL 查询, ATL 库。

### 2.1 ATL 模块

ATL 模块符合模型到模型的转换, 通过这种 ATL 单元能够制定从源模型元素产生目标模型元素的方法。

ATL 模块的源模型和目标模型必须被各自的元模型“typed”。下面介绍下 ATL 模型的结构。

1) Header section: 头部分定义了符合源模型和目标模型转换的模块名字和变量名字, 它也定义了模块的执行模型。有关头部分的语法定义如下:

```
module module_name;
```

```
create output_models [ from|refines] input_models.
```

2) Import section: 输入部分可以做选择性的输入, 它可以声明必须输入到哪个 ATL 库中。ATL 库声明式的实现如下:

```
uses extensionless_library_file_name.
```

3) Helpers: ATL 的 helpers 可以视为 java 里的方法。helpers 尽可能定义分解的 ATL 代码。一个 ATL helpers 用下面的元素定义: 一个名字, 一个文本类型, 一个返回值类型, 一个 ATL helpers 代码的表达式和一个可选的参数设置。例如, 下面是一个返回两个整数最大值的 helper 定义:

```
helper context Integer def : max(x : Integer) :
Integer = ...;
```

helpers 也可以定义一个不接受任何参数的 helper。下面是对一个乘以 2 的整数值的 helper 定义:

```
helper context Integer def : double() : Integer =
self * 2;
```

将以上二种定义混合的语法结构是:

```
helper def : max(x1 : Integer, x2 : Integer) :
Integer = ...。
```

4) Rules: 在 ATL 中存在二种不同类型的规则: 匹配规则和调用规则, 这二种规则符合通过 ATL 提供的二种不同的编程模型: 宣称性编程、命令式编程。

在 ATL 匹配规则中, ATL 转换的源模型元素不能被匹配, 详细设计源模型的匹配规则就是为了符合这种限制。另外, ATL 匹配规则不能生成 ATL 原始类型的值。调用规则可以被看作 helpers 的一个特定类型: 它们必须被明确地调用执行, 而且它们可以接受参数。然而, 相对于 helpers, 随着匹配规则, 调用规则可以生成目标模型元素。无论是从匹配规则或是其它的调用规则, 一个调用规则必须在一个命令代码部分中被调用。

### 2.2 ATL 查询

ATL 查询用于一个模型到原始类型值的转换, 它可以看作从一系列源模型计算原始值的操作。普遍使用的 ATL 查询是从一系列源模型中产

生一个文本输出。不过,ATL 查询不限于字符串值的计算,而且也可以返回数字值或是布尔值。下面分别介绍下 ATL 查询的结构和执行语义。

ATL 查询结构:完成一个可选的输入部分后,ATL 查询必须定义一个查询实例。关键字查询介绍了一个查询实例,查询实例指定了必须是由一个 ATL 表达式计算得到的结果。表达式如下:

```
query query_name = exp。
```

查询执行语义:作为 ATL 模块,ATL 查询的执行是在几个连续的阶段中实施的。第一阶段是初始化阶段,它对应着 ATL 模块的初始阶段,并认为是在 ATL 模块文本中定义的初始化属性。第二个阶段是计算阶段,通过执行 ATL 查询中查询元素的宣称代码去计算查询返回值。需注意的是,在查询文件中定义的 helpers 可以在初始化和计算阶段中被调用。

### 2.3 ATL 库

最后的 ATL 单元类型是 ATL 库,开发一个 ATL 库可以定义一系列的 ATL helpers,这些 helpers 可以被不同的 ATL 单元(包括模块、查询、库)调用。一个 ATL 库可以包括一个可选的输入部分,除此之外,ATL 库定义了一些 ATL helpers,这些 helpers 在导入库的 ATL 单元中可使用。相比模块和查询,一个 ATL 库不能被单独执行。这说明,库在执行时与其它初始化步骤不相关。由于缺少初始化的步骤,属性 helpers 不能在 ATL 库中定义。

### 3 模型转换

目前主流的 Atlas 转换框架<sup>[5]</sup>由 ATLAS 研究组、LINA & INRIA 和法国 Nantes 大学共同研究得出,如图 2 所示。框架中的源模型符合元模型 a,它只能读不能修改;目标模型符合元模型 b,它只能写。一个完整的 Atlas 模型转换程序需要四个文件:元模型 a、元模型 b、源模型、模型转换实例。源模型通过转换生成的目标是目标模型。

转换框架并没有详细的说明模型转换方法,文献[6]中的例子也只是把简单的类模型转换为关系模型,并未给出转换的详细过程,类的元模型及关系元模型如图 3、图 4 所示。目前在 MDA 的模型转换这一阶段,只是处于模型转换的理论性研究,模型转换技术尚未成熟。

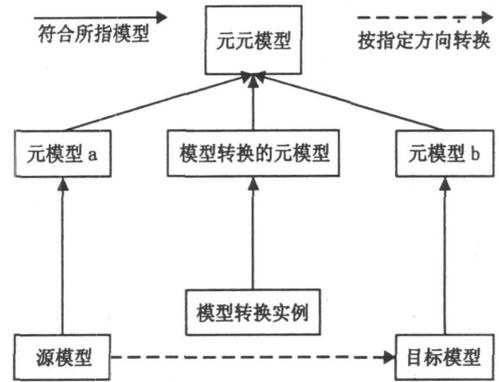


图2 Atlas转换框架  
Fig.2 Atlas conversion framework

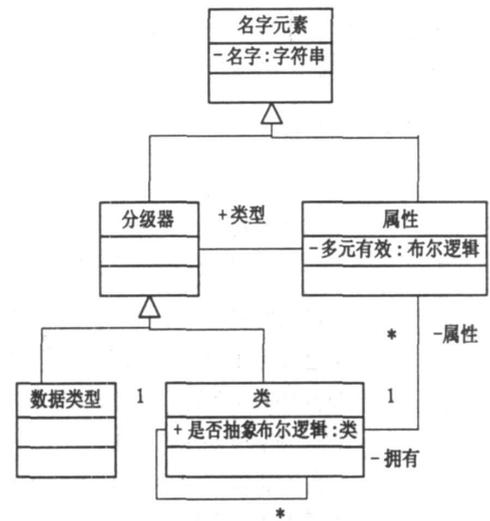


图3 类的元模型  
Fig.3 Class metamodel

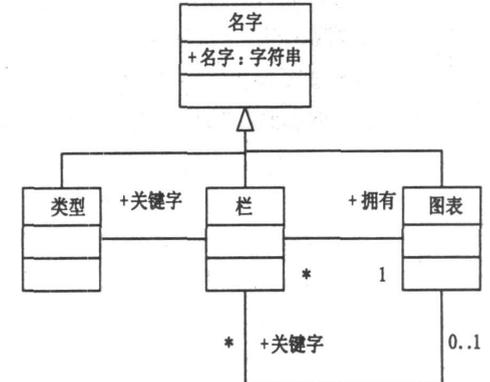


图4 关系元模型  
Fig.4 Relational metamodel

本文针对文献[6]及目前国内外的研究现状对模型转换进行了改进,提出一个利用 ATLAS 转换语言接收模型转换信号进行转换的思想框架。当 PIM 模型向 PSM 模型转换时,PIM 模型发

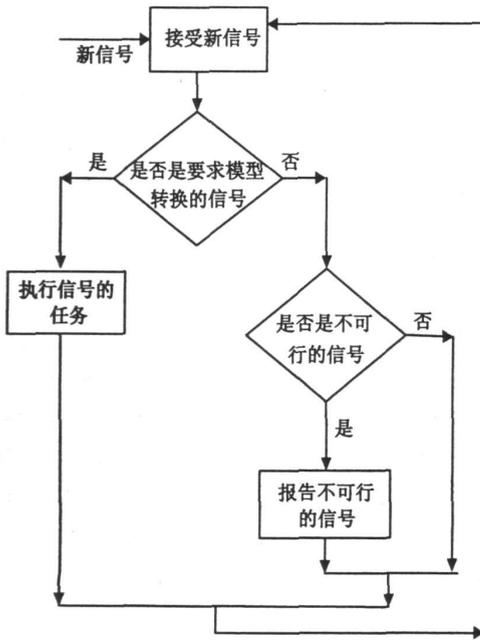


图5 利用ATL接受模型转换信号的思想框架

Fig.5 A framework of accepting model conversion signal with ATL

出一个要求模型转换的信号,ATL 接收到新信号时,就会判断此信号是否为模型转换的信号,如果是模型转换信号,则接受信号,并利用 ATL 执行此信号的任务,进行 PIM 模型到 PSM 模型转换,当完成任务后回到接收新信号的初始位置继续等待新的信号。如果 PIM 模型发出的信号不是 PIM 要求模型转换的信号,则进行判断该信号是否为可行的信号,如果是可行信号,则直接返回入口端等待新的模型转换信号;如果不是可行信号,则向数据库报告此信号为不可行信号,数据库内有不可行信号日志记录,将不可行信号写入日志是为方便

再次接收到类似信号时可在入口端直接屏蔽以免浪费循环时间,然后返回入口处等待接收新的信号任务,按此顺序循环下去,如图 5 所示。此循环过程的思想也可以应用在 PIM 到 PIM 的转换、PSM 到 PSM 的转换及 PSM 到 PIM 的转换中。

## 4 结语

理想的 MDA 模型转换是利用模型转换工具自动将 PIM 转换成相应的 PSM,但是目前支持 MDA 模型转换的工具无法做到自动转换。本文利用 ATLAS 转换语言,在 PIM 向 PSM 转换时判断接收的信号是否为转换信号,以便做出能否进行模型转换的决定。研究提出了一个模型转换的思想框架,为今后的模型开发提供借鉴与参考。

## 参考文献:

- [1] OMG/ORMSC. Model driven architecture. [EB/OL]. [2001-07-0]. <http://www.omg.org>.
- [2] 董建武. MDA: 新一代软件包互操作体系结构[J]. 计算机工程, 2003, 29(2): 3-4.
- [3] 赵珂, 韩业红. MDA 模型转换的研究[J]. 中国科技信息, 2009(4): 138-139.
- [4] JOUAULT F, KURTEV I. Transforming models with ATL [C]. Berlin: Proc. of the Satellite Events on the 8th Int'l conf. on Model Driven Engineering Languages and Systems, 2006(3): 128-138.
- [5] FRANKEL S. Model Driven Architecture: Applying MDA to Enterprise Computing [M]. Indianapolis: John Wiley and Sons, 2003.
- [6] JOUAU F, ALLILAIRE F, BEZIVIN J, et al. ATL: A model-transformation tool [J]. Science of Computer Programming, 2008(4): 31-39.

(责任编辑 马立)