

# 模型驱动下的数据库自动生成

王海林<sup>1,2</sup>

(1. 山西财经大学 信息管理学院, 山西 太原 030006;  
2. 奥塔哥大学 信息科学系, 奥塔哥 但尼丁 9054)

**摘 要:**为提高软件开发效率,提出模型驱动下的数据库自动生成方法。该方法以 MetaEdit+ 作为元建模工具,由领域专家建立领域元模型和模型,通过生成器定义语言 MERL,软件开发人员可以很方便地设计代码生成器,直接从领域专家所建立的图形领域模型生成 Java 程序代码,并运行已生成的程序代码进而生成数据库。通过一个实例详细介绍了数据库概念模型元模型设计、E-R 模型设计并给出从 E-R 模型到 Java 代码的生成器设计。经测试,所生成的 Java 程序代码可以在 Windows 操作系统环境的 Java 平台上运行,并能正确生成 Oracle 10g 数据库实例。

**关键词:**模型驱动; MetaEdit+; 元建模; 代码生成; 数据库生成

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2011)08-0173-04

## Automatic Generation of Database Based on Model Driven

WANG Hai-lin<sup>1,2</sup>

(1. Coll. of Inf. Management, Shanxi Univ. of Finance and Econo., Taiyuan 030006, China;  
2. Department of Information Science, University of Otago, Dunedin 9054, New Zealand)

**Abstract:** In order to promote software's development efficiency, is proposes an approach of database automatic generation based on model driven. This approach takes MetaEdit+ as a meta-modeling tool. Domain meta-models and models are established by domain experts. By using generator definition language MERL, the software developers can design code generators very conveniently. From the graph domain models which are established by domain experts, code generators can generate Java codes directly and then the generated codes can produce database. It introduces in detail design of database conceptual meta-model, E-R model and generators for Java code through an instance. The test result indicates that the generated Java codes can run correctly on Java platforms in Windows operating system environment and then can produce a database instance in Oracle 10g.

**Key words:** model driven; MetaEdit+; meta-modeling; code generation; database generation

## 0 引言

基于模型驱动架构 MDA (Model Driven Architecture)<sup>[1]</sup>的模型驱动开发方法 MDD (Model Driven Development)<sup>[1]</sup>是以模型为核心的软件开发方法。与基于通用建模语言 UML<sup>[2]</sup> (Unified Modeling Language) 的面向对象的开发方法不同, MDD 更注重应用领域, 更注重领域专家对建模过程的参与。也就是说, 它不要像 UML 这样的通用建模语言提供很多很全的功能, 只要求提供一种面向特定领域的建模语言 DSL (Domain Specific Language)<sup>[3]</sup>能快速有效地解决实际问题即可。由于 DSL 是面向特定领域的, 其提供的功能仅限于本领域且为领域专家所熟知, 所以简单易学, 整

个建模工作可由领域专家独自完成, 这样可以大大减轻软件开发人员的工作量, 使他们可以把更多的精力集中到程序代码生成器的设计上。文中所述的数据库自动生成方法实际上就是指利用已生成好的 Java 程序自动构建数据库的过程。

## 1 MetaEdit+

通常把设计 DSL 的工具软件称之为元建模工具 (meta-modeling tools), 为设计 DSL 所建的模型称为元模型 (meta-model), 而建立元模型和设计 DSL 的过程称为元建模 (meta-modeling)。由于领域专家对其所在领域更熟悉, 所以建立元模型、设计 DSL 以及建立模型的任务应该主要由领域专家来完成。因此在选择元建模工具时必须保证它简单易学, 这样只要对领域专家稍加培训他们即可掌握。目前元建模工具有很多, 比较著名的有 MetaEdit+、GME、DOME、EMF、GMF 等<sup>[4]</sup>, 从易用性和代码生成等因素考虑, 文中选择了

收稿日期: 2011-01-21; 修回日期: 2011-04-25

基金项目: 山西省省筹公派出国留学资金 (晋留管办发 2009-4 号)

作者简介: 王海林 (1962-), 男, 山西大同人, 副教授, 硕士, 研究方向为模型驱动软件开发、元建模、领域本体建模、XML 数据库等。

MetaEdit+作为元建模工具。

MetaEdit+是一款出现较早、使用最广的商用元建模工具,它提供了一组建模工具,如 diagram editors, matrix editor, table editor, browsers, report and code generation, method tools, API & XML connectivity, object repository 等<sup>[5]</sup>。使用 MetaEdit+可以大大缩短软件开发周期,软件开发效率提高 5 ~ 10 倍<sup>[6]</sup>,因为它提供了强大的代码生成功能,整个开发过程可不必编写代码<sup>[7]</sup>。下面文中以一个从数据库概念模型(即 E-R 图)到 Oracle 10g 数据库的生成过程为例详细介绍如何通过 MetaEdit+实现模型驱动下的数据库自动生成。

## 2 基于 MetaEdit+的元建模

在设计代码生成器之前,首先要建立领域模型。在 UML 中通常用类图对静态结构建模,用顺序图、状态图对动态行为建模<sup>[8]</sup>,而 MetaEdit+是一种元建模工具,它并没有事先将图分为类图、顺序图或状态图等,图的含义是由建模者自己定义。通过 MetaEdit+提供的工具建模者可以设计具有自己风格的图形符号和图,进而创建领域模型。

### 2.1 建立领域元模型

文中建立领域元模型是指设计数据库概念模型(即绘制出 E-R 图)以及程序代码生成所需的各种图形符号和元素。具体地讲,就是创建 MetaEdit+ 的 Property(属性)、Object(对象)、Relationship(联系)、Role(角色)和 Graph(图)。

#### 2.1.1 创建属性

通过 MetaEdit+ 的 Property Tool 工具创建每个对象的属性,创建的属性及说明如下:

Name [Software]:对象的名称

Index [Software]:对象在图中的顺序

Type:列的数据类型

ColumnWidth:列的宽度

DecimalDigit:Decimal 数据类型的小数位数

NotNull:列能否取空值

Unique:列是否取值唯一

PrimaryKey:列是否作为主键

Constraint:表中的用户定义完整性约束条件。它可以定义多个完整性约束条件,每个完整性约束条件为一个 Constraint 对象,一个 Constraint 对象由约束名和完整性约束条件组成。

图 1 给出了 Constraints 属性的设计界面。

#### 2.1.2 创建对象

元模型的对象实际上就是模型中的类(在数据库概念模型中为实体型),通过 Object Tool 工具创建。创建的对象有:

Constraint:表中的用户定义完整性约束条件对象

DBColumn:数据表的列对象,对应于 E-R 图实体型的属性

DBTable:数据表对象,对应于 E-R 图的实体型

图 2 给出了 DBTable 对象的设计界面。

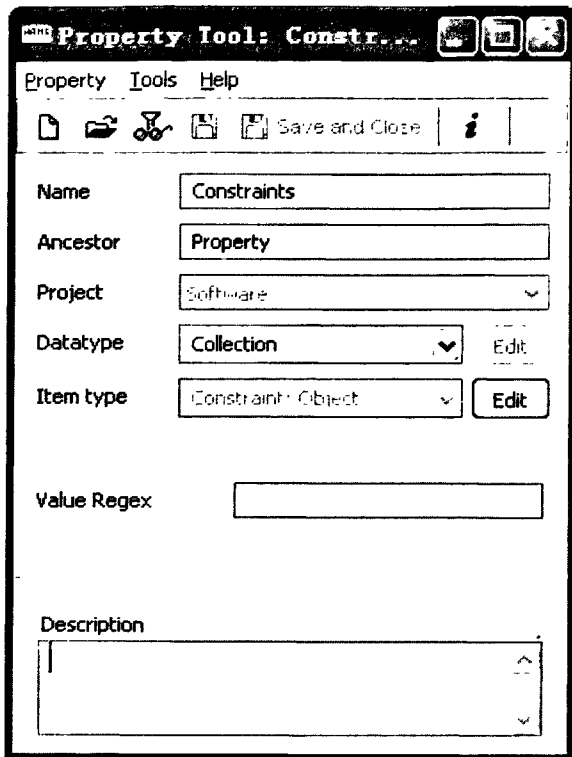


图 1 Constraints 属性的设计界面

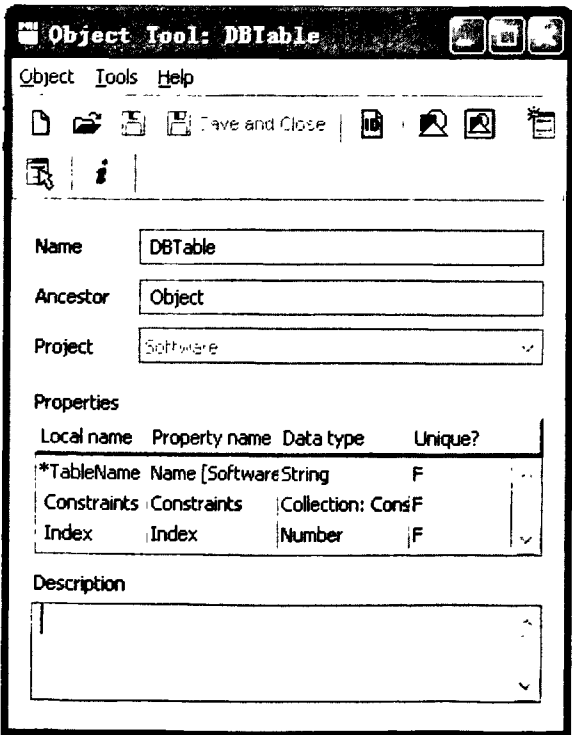


图 2 DBTable 对象的设计界面

#### 2.1.3 创建联系

使用 Relationship Tool 工具创建对象之间的联系

如下:

Contain:数据表与列之间的联系

Reference:数据表之间的联系,用来定义数据表之间的参照完整性。数据表之间的联系类型通常有三种,即1:1联系、1:n联系和m:n联系。m:n联系可用两个1:n联系来表示,而1:1联系又是1:n联系的特例(即n=1的情况),因此文中只创建了1:n联系类型,以简化E-R图的设计,其目的是为了使建模者(通常是领域专家)更容易掌握数据库概念模型的设计。

#### 2.1.4 创建角色

用Role Tool工具创建角色如下:

Table:联系中连接表对象的角色

Column:联系中连接列对象的角色

Parent:联系中连接父表对象的角色

Son:联系中连接子表对象的角色

#### 2.1.5 创建图

通过Graph Tool工具创建的图为:

ConceptualModel:可用于设计数据库概念模型的元模型图。

先将建好的对象、联系和角色添加到元模型图中,再将它们绑定。

### 2.2 建立领域模型

建立领域模型就是用上面所建元模型创建数据库概念模型,即绘制E-R图<sup>[9]</sup>。可通过Diagram Editor工具来设计数据库概念模型,图3给出了设计好的某Shopping System的E-R图。

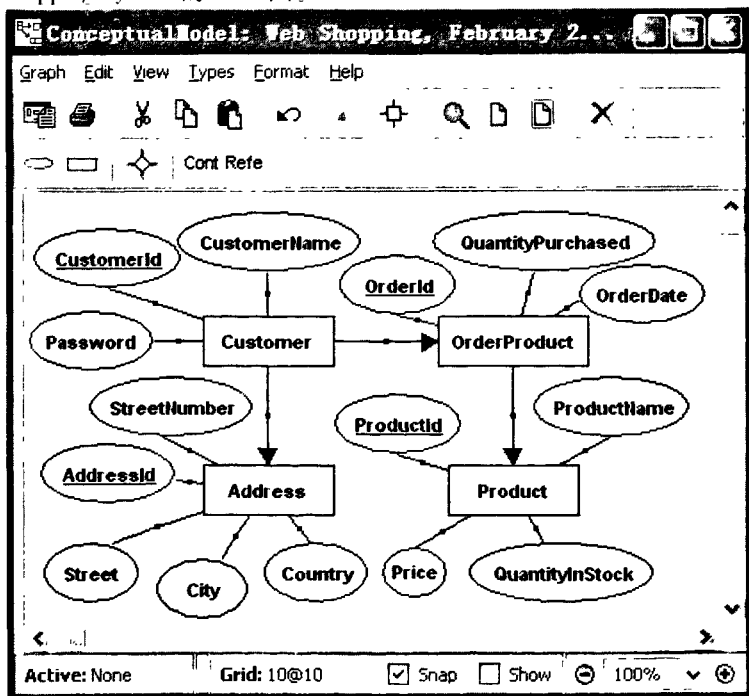


图3 某Shopping System的E-R图

图中属性名下方有下划线的属性为实体型的主键,带箭头的连线为1:n联系,带箭头的一方为n方。

为简化设计,图中没有标出父实体型(即联系中无箭头的一方)的外键,父实体型的外键可由生成程序根据子实体型(即联系中箭头所指的一方)的主键自动生成并添加到父实体型中。

### 3 代码生成器设计

模型建好后,接下来就是设计代码生成器将模型直接转换为Java程序进而生成数据库。MetaEdit+提供了功能强大的代码生成功能,包括生成器定义语言MERL和生成器设计器Generator Editor<sup>[10]</sup>。限于篇幅,下面只给出生成器\_TableCreate的设计代码及说明:

```
Report '_TableCreate' /* 生成器开始语句 */
foreach . DBTable; where :Index [ Software ] = '0' /*
* 循环语句,从E-R图Index [ Software ]值为0的对象
开始导航 */
/* 以下代码用于生成创建数据表的SQL语句 */
'sqlStr="create table ':TableName;('
do ~ Table>() ~ Column. DbColumn; orderby :Index [ Software ];
{ if :Index [ Software ] <> '1' then
',
endif
:ColumnName':Type
if (:Type = 'char') or (:Type =
decimal') then
('':ColumnWidth
if :Type = 'decimal' then
',':DecimalDigit
endif
endif
if :NotNull; then
'not null'
endif
if :Unique; then
'unique'
endif
}
do ~ Parent>() ~ Son. DBTable; orderby :Index [ Software ];
/* 循环语句,从父表导航到子表
```

```
do ~ Table>() ~ Column. DbColumn; where :Primary
Key; orderby :Index [ Software ];
```

```

/* 循环语句,为父表添加子表的主键,作为父
表的外键 */
',
:ColumnName':Type
if (:Type='char') or (:Type='decimal') then
'(:ColumnNameWidth
if :Type='decimal' then
',':DecimalDigit
endif
')'
endif
}
}
do ~ Table>() ~ Column. DbColumn; where :Pri-
maryKey; orderby :Index [ Software];
{ if :Index [ Software] = 1' then
',primary key(' /* 定义表的主键 */
else
',
endif
:ColumnName
}
')'
do :Constraints;
{ ',check(':Condition')' /* 定义表的用户定
义完整性约束条件 */
}')";newline
rs = c. getMetaData(). getTables( null, null, "' :
TableName% upper", null );
if (rs. next()) /* 判断表是否已经存在,若存在
则删除 */
st. executeUpdate( " drop table ':TableName% upper
");
}
st. executeUpdate(sqlStr); /* 用于执行创建表的
SQL 语句 */
'subreport '_TableCreates' run /* 调用子生成器,
继续创建其它表 */
}
foreach . DBTable; where :Index [ Software] = 0' /*
* 表添加参照完整性约束 */
{ do ~ Parent>() ~ Son. DBTable; orderby :In-
dex [ Software];
{ 'sqlStr = " alter table ':TableName;1;' add for-
eign key('
do ~ Table>() ~ Column. DbColumn; where :Pri-

```

```

maryKey; orderby :Index [ Software];
{ if :Index [ Software] <> 1' then
',
endif
:ColumnName
}
')'
'references ':TableName('
do ~ Table>() ~ Column. DbColumn; where :Pri-
maryKey; orderby :Index [ Software];
{ if :Index [ Software] <> 1' then
',
endif
:ColumnName
}
')'
";newline
'st. executeUpdate( sqlStr);
' }
subreport '_TableAddConstraints' run /* 调用子生
成器,继续为其它表添加约束 */
}
endreport /* 生成器结束语句 */
为了能在数据库中创建任意多个表,代码生成器
_TableCreates 和 _TableAddConstraints 均采用了递归调
用设计方式。

```

#### 4 结束语

用 MetaEdit+设计的代码生成器可以直接从 E-R 图形模型自动生成 Java 程序,进而由生成的 Java 程序生成数据库,所生成的程序可以在 Windows 环境的 Java 平台上正常运行,再由生成的 Java 程序创建了 Oracle 10g 数据库实例。通常 E-R 图中不会给出数据库名称,所以不能直接从 E-R 图动态创建数据库,只能由开发人员事先创建一个空数据库(数据库名称可根据应用领域来命名,文中为 shopping),由生成的 Java 程序自动连接该数据库<sup>[11]</sup>并创建数据库中包含的各种对象(主要是数据表)。文中介绍的方法具有普遍意义,用类似的方法还可以设计系统主窗口、多级菜单、各种功能模块<sup>[12]</sup>。总之,模型驱动下的代码自动生成方法可以大大提高软件开发速度,必将成为未来软件开发的主要方法。

#### 参考文献:

- [1] Richard S, David S F, John P. The MDA Journal: Model Driven  
(下转第 180 页)

应的整型数值,然后采用前文所述的整型压缩算法进行压缩存储。然后在地震数据的文件头参中做相应的标记,在解码时按照所述算法,将整型数据转为相应的浮点数即可。

### 3 试验结果

试验分别采用多个浮点型以及整型数据存储的 SEG-2 标准地震格式数据。试验证明,该方法无论对于整型道数据还是浮点型道数据,都可以对压缩后的数据实现无损还原。通过对压缩前后的波形图进行比较,可以看出无论是整型数据还是浮点型数据在压缩前后波形都无明显变化。另外,通过对大量文件压缩比的比较可发现:无论整型还是浮点型数据,其压缩比随着数据波动范围的增大而增大,同时,随着波动强度的增加而适当地减小。

据试验所得数据中,最高压缩比可达 3.6 : 1,最小压缩比可达 1.3 : 1。这说明,该方法无论是对浮点型还是整型数据进行压缩时,对于波动整体数值较大的波形,或者,相邻点变化较慢的波形压缩明显;对于整体数据很小,而且相邻点变化较大的波形压缩效果较弱。

### 4 结束语

采用文中描述方法对地震数据进行压缩存储,将浮点型数据与整型数据区分对待,最大限度地提高了数据无损压缩的比例。无论整型数据还是浮点型数据,其解压缩的结果都是完全无损的,保证了数据的精确性。通过试验结果分析,可以得出压缩算法理想的

适用范围。

#### 参考文献:

- [1] 冯占林, 张学工, 李衍达. 基于小波变换的地震勘探数据压缩的工程分析[J]. 清华大学学报(自然科学版), 2001, 41(4): 170-173.
- [2] 王国清. 小波变换理论及其在地震勘探数据压缩中的应用[D]. 南京: 南京理工大学, 2005.
- [3] 孙寰宇, 武文波, 杨志高, 等. 基于 JPEG2000 的地震数据无损压缩[J]. 计算机工程与应用, 2005, 41(16): 185-188.
- [4] Society of Exploration Geophysicist SEG Y rev 1 Data Exchange Format[S]. 2003.
- [5] SEG 委员会. SEG D Rev2, SEG Field Tape Standards[S]. 1996.
- [6] 王秀文, 姚立平, 赖德伦, 等. 地震数据交换标准[J]. 地震地磁观测与研究, 1994, 2(15): 1-42.
- [7] 罗新恒, 正哲, 王春明, 等. SEED 数据压缩率的比较[J]. 地震地磁观测与研究, 2004, 24(4): 41-47, 14-19.
- [8] Swedens W. The lifting scheme: a new philosophy in biorthogonal wavelet constructions[J]. Proceedings of SPIE, 1995, 2569: 68-79.
- [9] 王文涛. 基于小波的图像压缩编码算法研究[D]. 重庆: 重庆大学, 2005: 22-24.
- [10] Mandyam G, Magotra N, McCoy W. Lossless Seismic Data Compression using[M]. [s. l.]: Adaptive Linear Prediction, 1996.
- [11] ISO/IEC JTC1/SC29/WG1. Coding of still pictures[S]. 2000-03-16/2005-04-11.
- [12] 孙毅. 浮点型数据的无损压缩[C]//ICRCCS. 常州: [出版者不详], 2010: 24-27.

(上接第 176 页)

- Architecture Straight from the Masters[M]. New York: Meghan Kiffer Press, 2004: 135-146.
- [2] 谢正良, 赵建华, 李宜东, 等. 一种基于 J2EE 平台的 MDA 模型转换技术[J]. 计算机应用研究, 2005(3): 51-54.
- [3] Martin F. Domain Specific Languages[M]. Toronto: Addison-Wesley Professional, 2010.
- [4] 刘辉, 麻志毅, 邯维忠. 元建模技术研究进展[J]. 软件学报, 2008, 19(6): 1317-1327.
- [5] 周金根. MetaModelEngine: 元模型引擎开发思路[EB/OL]. [2010-10-12]. <http://www.cnblogs.com/zhoujg/archive/2010/07/28/1786155.html>.
- [6] MetaCase. MetaEdit+Domain-Specific Modeling Environment[EB/OL]. [2010-09-10]. <http://www.metacase.com/MetaEdit.html>.
- [7] Juha-Pekka T. MetaEdit+: integrated modeling and meta-modeling environment for domain-specific languages[C]//Companion to the 21st ACM SIGPLAN symposium on object-oriented programming systems, languages, and applications. New York: ACM, 2003: 690-691.
- [8] 李思广, 林子禹, 胡峰, 等. 基于 UML 的软件过程建模方法研究[J]. 计算机工程与应用, 2003, 39(6): 76-78.
- [9] 王珊, 萨师焯. 数据库系统概论[M]. 第 4 版. 北京: 高等教育出版社, 2006.
- [10] MetaCase. MetaEdit+4.5 Workbench User's Guide[EB/OL]. [2010-09-15]. <http://www.metacase.com/support/45/manuals/mwb/Mw.html>.
- [11] 周彩兰, 孙琳, 李素芬. 基于 JSP 的网络数据库连接技术[J]. 计算机技术与发展, 2006, 16(4): 209-211.
- [12] Sanna S. Domain-specific modeling language and code generator for developing repository-based Eclipse plug-ins[R]. Espoo: VTT Publications, 2008.

作者: 王海林, WANG Hai-lin  
作者单位: 山西财经大学信息管理学院, 山西太原030006; 奥塔哥大学信息科学系, 奥塔哥但尼丁9054  
刊名: 计算机技术与发展 ISTIC  
英文刊名: Computer Technology and Development  
年, 卷(期): 2011, 21 (8)

参考文献(12条)

1. Richard S;David S F;John P [The MDA Journal:Model Driven Architecture Straight from the Masters](#) 2004
2. 谢正良;赵建华;李宣东 [一种基于J2EE平台的MDA模型转换技术](#)[期刊论文]-[计算机应用研究](#) 2005 (03)
3. Martin F [Domain Specific Languages](#) 2010
4. 刘辉;麻志毅;邯维忠 [元建模技术研究进展](#)[期刊论文]-[软件学报](#) 2008 (06)
5. 周金根 [MetaModelEngine:元模型引擎开发思路](#) 2010
6. MetaCase [MetaEdit+Domain-Specific Modeling Environment](#) 2010
7. Juha-Pekka T [MetaEdit +:integrated modeling and metamodeling environment for domain-specific languages](#) 2003
8. 李思广;林子禹;胡峰 [基于UML的软件过程建模方法研究](#)[期刊论文]-[计算机工程与应用](#) 2003 (06)
9. 王珊;萨师煊 [数据库系统概论](#) 2006
10. MetaCase [MetaEdit+4.5 Workbench User' s Guide](#) 2010
11. 周彩兰;孙琳;李素芬 [基于JSP的网络数据库连接技术](#)[期刊论文]-[计算机技术与发展](#) 2006 (04)
12. Sanna S [Domain-specific modeling language and code generator for developing repository-based Eclipse plug-ins](#) 2008

引用本文格式: 王海林, WANG Hai-lin [模型驱动下的数据库自动生成](#)[期刊论文]-[计算机技术与发展](#) 2011 (8)