维普资讯 http://www.cqvip.com

计算机科学 1995Vol. 22№. 1

从关系模式到面向对象模式的转换

吴胜利 王能斌
（东南大学计算机系 南京 210018）

摘 要 This paper discusses the transformation problem from relational schema to object-oriented schema, which is one of the important aspect of heterogeneous federated database system, and object-oriented data model is a good candidate for global data model.

关键词 Schema transformation, Schema translation, Relational schema, Object-oriented schema.

异构联邦数据库系统研究将涉及模式转换和模式集成两个关系密切的重要方面。八十年代中期以后，人们对模式集成问题更感兴趣。一般认为，面向对象模型是实现全局数据模型的较佳选择[1]，其语义功能较强，转换比较容易实现，而且转换的结果只是作为模式集成的输入。尤其是关系模式，不作任何变动就可成为面向对象模式。尽管'模式'未反映面向对象模型的任何特点。因此，关键在于开发一个功能非常强大的模式集成器。然而像这样的模式集成器是很难实现的。另外，许多联邦数据库系统并无全局模式，其模式转换所得到的是最终结果，但这时不应以此为满足，而应进行优化。

八十年代中期以前对模式转换研究较多，主要是关系、层次、网状和实体-联系诸种模式间的转换。目前对面向对象与其它模式之间的转换研究很少。我们认为，这方面的研究还是有价值的。本文将讨论从关系模式到面向对象模式的转换。

1. 关系模型

我们限定关系模式满足第三范式标准，可简化转换过程。这样的限定可以被接受，一是大多数用户设计的数据库能满足或稍作修改就能满足要求，二是多年前就有人提出了将第一范式分解为第三范式的算法[2]。

每一个关系由以下五项组成：关系名、属性集、主码、外部码和各主属性联系（如果主码含多个属性）。

下面以两属性主码为例解释一下多属性主码的各主属性联系。若一关系主码包括 A，B 两属性，且对该关系任意可能的元组，一个 A 属性值可和一个 B 属性值成为码值（反之亦然），则说 A，B 两主属性为一一对应联系（记作 1∶1）；若一个 A 属性值可和多个 B 属性值组成码值，则 A 与 B 为一对多联系（记作 1∶n）；若一个 A 属性值可和多个 B 属性值组成码值，且一个 B 属性值可和多个 A 属性值组成码值，则 A 与 B 为多对多联系（记作 n∶n）。不难将这种联系推广到一般情形。

此外我们假定在关系模式中，属性的命名是统一的，亦即一特定属性名在所有关系中均具有相同的语义解释。

2. 面向对象模型

所有的对象均由系统定义的标识符唯一标识。该模型支持类型间的类属和分属联系，支持集合概念，集合可涉及一个或多个属性。例如，可如下定义学生类型：

```
DEFINE TYPE 学生
ATTRIBUTE 学号 INTEGER(6)
          姓名 CHAR(10)
          系别 CHAR(10)
          {课程 CHAR(10),
           成绩 INTEGER(3)};
```

这里{ }表示取其中诸属性值的集合。

'学生'类型的两个可能的实例对象如图 1 所示：

学号	姓名	系别	{课程	成绩}
S01	王英	数学	C01 C02 C03	80 85 85
S02	张立	数学	C01 C02 C03	85 90 80

图 1 '学生'类型的两个实例对象

维普资讯 http://www.cqvip.com

3. 关系(型)之间的联系

与面向对象模型支持较多的语义联系不同，关系模型只支持很少的语义，其大多数语义联系只是存贮在数据库设计者的脑海中，不能在模式中显式表现出来。为得到一个较优的转化，应充分挖掘关系数据库模式中的潜在语义联系。

由于很少涉及到关系实例的概念，故本文以下将关系型(或关系框架)简称为关系。此外，我们认为在现实世界中必定有一特定的实体类或联系类与一关系相对应。若两关系 R_1、R_2 描述的是相同的实体类或联系类，则说 R_1 和 R_2 有相同的语义解释。若关系 R_1 所描述的实体类或联系类是 R_2 所描述的实体类或联系类的子集，则说 R_1 是 R_2 语义解释的特殊化，而 R_2 是 R_1 语义解释的一般化。R_1 是 R_2 的子类，R_2 是 R_1 的超类。

下面列出关系间可能满足的一些条件，并讨论在这些条件下关系间可能具有的联系：

a)关系 R_1 与 R_2 除其关系名外完全相同。

b)关系 R_1 的全部属性集为 R_2 的子集，且 R_1 和 R_2 的主码及主属性联系相同。

c)关系 R_1 与 R_2 主码及主属性联系相同，但其它属性无相同者。

d)关系 R_1 与 R_2 主码及主属性联系相同，但其它属性不全相同，且不为 b)所述情形。

e)关系 R_1 的主码为另一关系 R_2 的属性(集)。

f)关系 R_1 有多个主属性，其中一些(或全部)主属性是别的关系中的属性，另一些(或空)主属性是 R_1 中特有的。

若 a)成立，则 R_1 与 R_2 可能具有相同的语义解释，这时可将 R_1 与 R_2 合并为一。但更可能 R_1、R_2 具有不同的语义解释，不能合并。可以肯定的是，R_1 与 R_2 具有某种紧密的联系，例如可能具有类属联系(R_1 为'学生'，R_2 为'优秀学生')，或为同一超类的两个子类(R_1 为'男学生'，R_2 为'女学生')。

若 b)成立，则 R_1 为 R_2 的超类，R_2 为 R_1 的子类。

若 c)成立，则 R_1 和 R_2 可能是描述同一实体型或联系型的不同方面，这时 R_1 和 R_2 可以合并。也可能 R_1 和 R_2 具有某种紧密联系，如它们中一个是另一个的子类。

若 d)成立，则 R_1 和 R_2 肯定不具有类属联系。R_1 和 R_2 可为同一实体型或联系型的不同方面(这时有冗余存在)，或 R_1 和 R_2 为同一超类的两个子类。

若 e)成立，则 R_2 与 R_1 具有分属联系。

若 f)成立，则 R_1 关系描述的是联系型，主属性的联系方式即是实体型之间联系方式的体现。若 R_1 中存在非主属性，则为联系属性。

由以上可知，根据关系模式决定类属联系比较困难，而分属联系则比较明确。因此在转换过程中，需要人工干预确定类型间的类属联系。

4. 转换过程

输入 一个关系模式中所有关系。每一关系包括关系名、属性集、主码、外部码和主属性联系方式。

输出 一个面向对象数据库模式。

具体步骤

1. 对关系模式中每一关系，执行以下第 2 和第 3 步；

2. 对关系 R_1，若不存在另一关系 R_2，使得 R_1 中全部属性为 R_2 的子集，则在面向对象模式中生成一类型 R_1，为根类型的直接子类；

3. 若关系 R 的全部属性为类型 $R_1, R_2, \cdots, R_n$ 所含属性的真子集，且 $R, R_1, R_2, \cdots, R_n$ 具有相同的主码和主属性联系，则定义 R 为 $R'_1, R'_2, \cdots, R'_j (j \leqslant i)$ 的直接子类。其中 $R'_1, R'_2, \cdots, R'_j$ 是 $R_1, R_2, \cdots, R_n$ 的子集，并且 $R'_1, R'_2, \cdots, R'_j$ 中属性的真子集均不能包含 $R_1, R_2, \cdots, R_n$ 中任一个关系的全部属性(因为 $R_1, R_2, \cdots, R_n$ 中的一些关系亦可具有类属联系)；

4. 重复以上两步，直至所有关系均被转换；

5. 由操作人员对满足条件 a)、c)、d)的关系增加类属联系或合并类型；

6. 对关系模式中每一关系进行第 7 至第 9 步操作(若在第 5 步进行了类型合并，则应将合并前后类型信息保留在此应用)；

7. 若关系 R 中属性 A 为另一关系 R_1 的主码，则定义 A 的值域为类型 R_1，即定义 R_1 为 R 的组成类型。更一般地，若 A 同时为 $R_1, R_2, \cdots, R_n$ n 个类型的主码，则找到 $R_1, R_2, \cdots, R_n$ 的最近共同祖先 R'，定义 A 的值域为 R'；

8. 若关系 R 中属性 A 为另一关系 R_1 的主属性，R_1 有多个主属性，R_1 中主属性 A 所对应的主属性联系为 1*)，则可撤销 R_1 类型，将 R_1 类型的全部属性(A 除外)并入 R 类型。

9. 若关系 R 的主码有多个属性，其中一些主属

*)在 R_1 中，假定主属性为 $A_1, A_2, \cdots, A_n$，其主属性联系为 $t_1:t_2:\cdots:t_n$，此时 A_1 所对应的 t_1 为 1，而无论 $t_2, \cdots, t_n$ 为 1 或 n。

维普资讯 http://www.cqvip.com

性分别是关系 $R_1, R_2, \cdots, R_n$ 的主码，而另一些（可空）为 R 特有的主属性，且 $A_1, A_2, \cdots, A_n$ 所对应的主属性联系方式为 n（此时不满足第 8 步的条件），则定义 R 中 $A_1, A_2, \cdots, A_n$ 的值域分别为 $R_1, R_2, \cdots, R_n$。亦即定义 $R_1, R_2, \cdots, R_n$ 为 R 的组成对象；

10. 重复第 7 至第 9 步，直至所有关系处理完毕，结束。

该转换过程主要分两阶段进行。第一阶段为第 1 至第 5 步，生成面向对象模式中的类型和类属联系，第二阶段为第 6 至第 10 步，生成类型间的分属联系。

需要说明一下标识符。面向对象模型支持系统定义的标识符，而关系模型只支持主码概念。如果进行数据库系统的转换，并没有什么问题，但在以面向对象模型作全局模型的联邦数据库系统中，只进行模式转换，数据库实例是不存放在全局模式中的，此时需要考虑标识符问题。关键是需要在面向对象模型中的标识符和关系模型的码值之间建立一种一一对应联系，如可定义面向对象模型中的标识符为类型信息加上相应关系中元组的主码值。

对关系视图我们亦未加以讨论，由于面向对象模型可以方便地定义视图机制[3]，故亦不难转换。

5. 一个例子

假设有一关系数据库模式包含以下关系：

教职工{职工号，主码/姓名/年龄}，
学生{学号，主码/姓名/年龄}，
优秀学生{学号，主码/简介}，
在职学生{职工号，主码 1/学号，主码 2/姓名/年龄}，
课程{课程号，主码/课程名/任课教师}，
学生成绩{学号，主属性/课程号，主属性/成绩}；

采用上一节介绍的转换过程进行转换后，得到如图 2 所示的面向对象模式。其中实践表示类属联系，虚线表示分属联系。另外需指出的是'学生'与'优秀学生'之间的类属关系需操作人员指定。

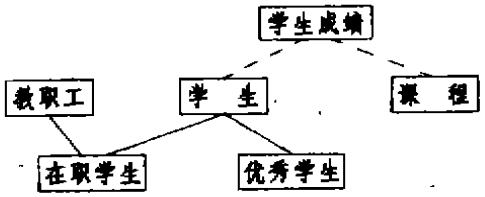

图 2 转换后所得的面向对象模式

由于本文只限于讨论模式转换，故未涉及实例转换的有关内容。

参考文献

[1]F. Salter, et al., Suitability of Data Models as Canonical Models for Federated Databases, SIGMOD RECORD, Vol. 20, No. 4, Dec., 1991

[2]B. Salzberg, Third Normal Form Made Easy, SIGMOD RECORD, Vol. 15, No. 4, Dec., 1986

[3]吴胜利，对象的视图与外模式，计算机研究与发展，1994 年第 8 期

[4]A. Motro, Superviews: Virtual Integration of Multiple Database, Transaction of Software Engineering, Vol. 13, No. 7, July 1987

（上接第 42 页）

[11]H. B. M. Jonkers, Inheritance in COLD, Algebraic methods I, Theory, tools and applications, Berlin: Springer-Verlag, 1991, pp. 277-302

[12]金娟哲，金淳兆，面向对象软件规格语言的设计，软件学报，待发表。

[13]Ian Hayes, Specfication case studies, Prentice-Hall Intl. (UK)Ltd, 1987

[14]C. Stanley-Smith, et al., UNIFORM, a language geared to system description and transformation, University of Limerick, Ireland, 1990

[15]金娟哲、余江、金淳兆，可重用构件及其描述语言，软件学报，vol. 5 no. 1, 1994, pp. 42-46

[16]B. Henderson-Sellers et al., The Object-Oriented System Life Cycle, CACM, 1990, 33(9): 142-159

[17]B. Henderson-Sellers et al., The O-O-O methodology for the object-oriented life cycle, ACM SIGSOFT Soft. Eng. Notes, 1993, 18(4): 54-60

[18]L. R. Hodge et al., A proposed object-oriented development methodology, Soft. Eng. J., March 1992 pp. 119-129