对象 / 关系映射技术与面向对象数据库技术比较分析

梁文菲[1] 黄厚宽[2]
1、北京交通大学软件学院
2、北京交通大学计算机学院

摘 要
本文通过对象 / 关系映射技术与面向对象数据库技术比较分析阐述了这两种技术的优缺点以及这两种技术的发展趋势。

关键词
对象 / 关系映射;面向对象数据库

Abstract
By O / R mapping technology and object-oriented database technology comparative analysis of the advantages and disadvantages, this text described bothtechnologytrends.

Key words
ORM;OODB

引言

关系型数据库发展至今已经相当成熟，而随着应用复杂性的提高，关系数据库已经越来越难以满足现实的需要。这主要表现在它的效率性能、可扩展性、使用的简洁性以及和当今开发技术的适应性。尽管表格结构的简洁可以支持强大的查询语言（SQL），但它不属于面向对象领域，现实世界的数据很难分解为这种简单的行列结构。这就是所谓“阻抗失谐（Impedance Mismatch）”问题。为了解决这个问题，面向对象技术和数据库技术自然而然开始交流和结合，并开始影响未来数据库的发展。

从目前来讲，对付“阻抗失谐”问题大体上有两种方法：对象 / 关系映射技术与面向对象数据库技术。

一、对象 / 关系映射技术与面向对象数据库技术简介

对象 - 关系映射 (Object/Relation Mapping，简称ORM)，是随着面向对象的软件开发方法发展而产生的。用来把对象模型表示的对象映射到基于SQL 的关系模型数据库结构中去。这样，我们在具体的操作实体对象的时候，就不需要再去和复杂的SQL 语句打交道，只需简单的操作实体对象的属性和方法。

面向对象数据库技术是面向对象技术与数据库技术相结合的产物，产生于 20 世纪 80 年代后期。它利用类来描述复杂对象，利用类中封装的方法来模拟对象的复杂行为，利用继承性来实现对象的结构和方法的重用。这种系统最显著的特点就是数据与代码不再是相互独

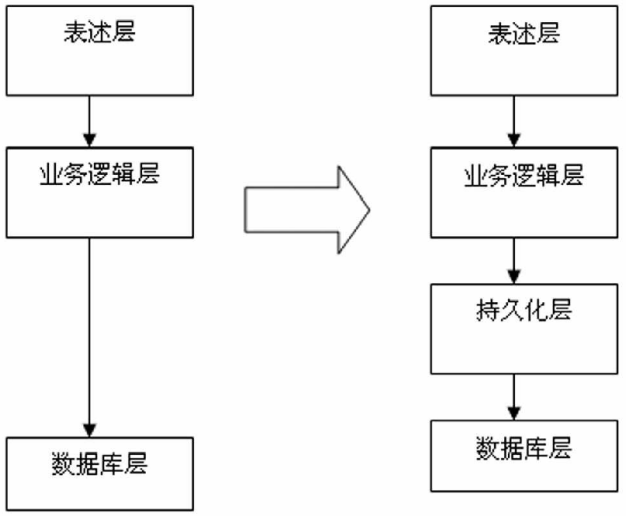

图 1

立的，对数据的运算必须通过调用定义于其上的操作即函数来实现。

从上面我们可以发现：ORM 技术是在对象和关系之间提供了一条桥梁，前台的对象型数据和数据库中的关系型的数据通过这个桥梁来相互转化；而 OODB 技术则是直接将关系型数据库放弃，代之以新类型的数据库。

从概念上看，ORM 技术并没有从本质上解决“阻抗失谐”问题，因为在数据库设计阶段仍然要将对象拆分成一个个的表格，ORM 技术只是把它掩盖了起来，在程序员这个层次解决了“阻抗失谐”问题；而 OODB 技术显而易见从根本上解决了“阻抗失谐”问题。从表面上看，ORM 技术似乎劣于 OODB 技术，因为前后台通过一个“桥梁”进行对话总没有直接对话方便。但是在实际的应用中，ORM 技术的应用目前要远远比 OODB 技术的应用流行。下面我们通过分析这两个技术的优缺点来探讨其原因。

二、对象 / 关系映射技术与面向对象数据库技术优缺点

2.1、对象 / 关系映射技术优缺点

优点：

(1) 提高性能

通过 Cache 的实现，能够对性能进行调优，实现了 ORM 区隔了实际数据存储层和业务逻辑层之间的关系，能够对每一层进行单独跟踪，增加了性能优化的可能。如图 1 所示。

(2) 能够以面向对象的方式操纵数据

可以直接处理业务对象，我们根本不用操心 SQL 语句以及底层存储方式，这样极大地简化了代码，提高了开发效率，对于日后的维护以及扩展都带来极大的便利。

但是，这并不代表完全抛弃 SQL，实际上，对于复杂的查询和报表，仍然推荐使用 SQL 方式访问数据库。ORM 的好处是使我们多了选择的余地。

以 hibernate 为例，用下列代码简单描述一下 ORM 的方便之处：

```
Session session = getSessionFactory().openSession();
    Transaction tx = session.beginTransaction();
    classA a = new classA ();
    ……….    // 对对象 a 的操作
    session.save(message);
    tx.commit();
    session.close();
```

classA 是一个持久化类，用它生成的对象的数据经过处理后需存储到数据库中，它对应一个或多个数据库表。我们只要建立一个名为 classA.hbm.xml 的配置文件，该文件建立类 classA 与数据库表的对应关系。在配置好毕要的代码环境之后，就可以像上面的代码那样操作数据库。

代码中斜体加粗部分即为将对象 a 的相关数据存储到数据库的操作，其它代码为环境的配置，我们不必关心。整个过程中没有一句 SQL 语句，这样就使我们以往频繁的 SQL 中摆脱出来，专心处理业务领域的问题。

(3) 隔离数据源，可以很方便地转换数据库

由于 ORM 将业务层与数据库层分开，程序员不用关心实际存储的方式，如果我们需要把后台数据库由 SQL Server 数据库换为 ORACLE 数据库，按照传统方式需要做大规模的修改，而使用 ORM 则只需要修改相应的配置文件而不需要修改程序。

缺点：

(1) 屏蔽了底层使得我们无法针对具体数据源做优化

关系数据库有着三十年的发展经验，其对数据源的优化远不是 ORM 能够相比的，因此，ORM 简化了数据库访问的同时，使得我们不能像优化 SQL 一样来优化 ORM，这样在性能方面会受到影响，尤其是在应用规模逐渐膨胀的情况下。

(2) 缓存策略存在缺陷

条件查询的时候，如果 Query Key 已经存在于缓存，那么不需要再查询数据库。但是，如果有任何一条记录发生变化，那么缓存中所有和该表相关的 Query Key 都会失效。

比如，有这样几组 Query Key，它们的 SQL 里面都包括 table1。

```
SQL = select * from table1 where a = ?, and b= 1
SQL = select * from table1 where a = ?, and b = 1
SQL = select * from table1 where a = ?, and b = 2
SQL = select * from table1 where a = ?, and b = 2
```

当table1的任何记录发生改变的时候，这些 Query Key 对应的结果集都不能保证没有发生变化。也就是说，很难做到根据数据的改动精确判断哪些 Query Key 对应的结果集受到影响。

(3) 没有从本质上解决“阻抗失谐”问题

ORM 对于复杂关联的 SQL 操作有些力不从心，因此利用 ORM 技术来映射，会有许多数据无法很好地映射到数据表上，这是 OR 的本质差异造成的。由此带来的结果是：从对象模型出发的时候，还要同时考虑关系数据库的东西，加入 ORM 技术之后，还得考虑当前使用的 ORM 工具的特性，如果 OR 之间差异很大的话，就需要对象模型照顾关系数据模型——不是对象驱动开发，而是数据库驱动开发——显然这是一种比较别扭的 OO 设计，给人本末倒置的感觉。

上述缺点本质上还是由“阻抗失谐”造成的：数据模型的不匹配，使得必须使用一种额外的方法来进行转化，不管这种方法是方便快捷的（ORM）还是不方便快捷的（SQL），都是不属于应用的业务逻辑方面的。并且，两者之间也难以达到完全准确的转换。就好比两个操不同语言的人进行交流，尽管有翻译来帮助交流，但是终究没有双方只用一种语言进行交流表达得透彻。

2.2、面向对象数据库技术优缺点

优点：

(1) 从根本上解决了“阻抗失谐”问题

面向对象设计是符合人类的思维习惯的，不必把对象数据转换成关系数据，因此设计过程更加准确，不容易出现大的设计偏差，封装性和信息隐藏技术提供了程序的模块化机制，继承和类层次技术提供了软件的重用机

制 。

(2) 可维护性好

在耦合性和内聚性方面，面向对象数据库的性能尤为突出。这使得数据库设计者可在尽可能少影响现存代码和数据的条件下修改数据库结构。这种先进的耦合性和内聚性也简化了在基于不同平台的网络分布式数据库的运行 。

对数据库能够进行对象化的查询，是对数据库进行彻底的面向对象化。这体现在我们使用一种全新的数据库查询语言——对象查询语言（OQL），它能够简洁易懂地对数据库中的对象进行查询。OQL查询和SQL查询类似，但是使用的是对象名称而不是SQL名称，并且不需要join子句。举例如下 ：

假设有两个数据类：Customer和Order，它们之间有一对多的关系：Order.Customer和Customer.orders。在数据库中已经存在很多这两个类的实例，以及相互之间的关系。我们可以使用下面的对象式查询语言来查询符合条件的User对象：

```
select * from Order where
Order.Custom.name=
'Customer1'。
```

从中我们可以看出，通过使用面向对象中的成员属性指定符“.”，可以让我们达到SQL中的连表的效果，实际上，第一个句查询的SQL等价版本是 ：

```
select a.* from Order a,
Customer b
where a.CustomID = b.ID
and b.NAME =
'Customer1'
```

由此可见，OQL比SQL语言简单并且表意明确，更符合使用者的思维习惯。在类图比较复杂、查询涉及的类又比较多的时候，OQL会体现出它的优势。

(2)节省存储空间

以对象方式存储与程序员使用面向对象方法对数据的表述一致，避免了以往在转换数据模型时产生过多的数据冗余，因而节省了存储空间。

缺点 ：

(1) 没有统一的模式和统一的标准语言

大多数OODB都用自己独特的方法实现各种特色功能。但各个OODB各自为政却带来了一些麻烦，要找出一个完全顺从ODMG 2.0或3.0规范（ODMG，Object Database Management Group，对象数据库管理组织）的OODB产品很困难。OODB比起RDB来，不只是基本的几种数据类型那么简单，它还涉及继承处理、多态等一大堆面向对象特征的实现。

(2) 它的数据模型并不是建立在完善的数学基础之上，数据库语言缺乏形式化基础。

实际上这条不算是严重的缺陷，因为关系数据库有一个统一标准，但是代价是放弃了现实世界的多样性。

(3) 不与SQL 标准兼容，难以同现有的关系数据库进行有效的转换

这个缺点是OODB的致命伤，因为SQL 标准是数据库操作最流行的接口，而OODB使用的OQL虽然表意明确，但是在查询速度上远不如SQL。因此与SQL 标准的兼容性成为制约OODB发展的瓶颈。另一方面，从商业因素上考虑，现在主流的数据库技术仍然是关系数据库技术，绝大部分应用是建立在关系数据库基础之上的，若试图说服企业使用OODBMS，必须能够将历史数据准确无误地转移过来，而这正是难以完成的。因此，OODB目前也只有在一些特定的对查询要求不高的非事务性应用领域，如计算机辅助设计(CAD)，地理信息系统（GIS）等领域，才能发挥自己的优势。这也是OODB自80年代发展至今一直没有得到广泛应用的根结所在。

三、分析总结

从以上ORM与OODB的优缺点可以做出以下总结：

由于OR本质上的差异，导致两者之间有“阻抗失谐”这样一条不可逾越的鸿沟，随着应用复杂性的提高，这个鸿沟会越来越明显，ORM可以暂时缓解这个矛盾但不可以根本解决这个矛盾，至少从目前看来如此。OODB技术能从根本上克服关系数据库的缺点，但是受性能影响难以在短期内大规模地使用。因此，在现阶段ORM相比OODB得到了更广泛的应用，作为关系数据库的有益补充。

从这两种技术的发展趋势来看，ORM的存在依赖于关系数据库，因此只要关系数据库没有被面向对象数据库完全取代，ORM技术是会一直发展下去的。实际上，这涉及到OODB和RDB的关系问题，现在人们普遍认为，两者不是RDB与网状层次数据库那样的取代关系，两者是共同发展的关系。

面向对象数据库的发展主要看它怎样克服其技术瓶颈，即与SQL标准的兼容性方面。从上面的分析看出，纯的面向对象数据库技术似乎走向了死胡同，因此可以考虑将关系的元素加入到其中，即所谓的对象-关系数据库，这也是面向对象数据库的发展方向之一，它的好处是底层实现不用像纯的面向对象数据库那样从头做起，而是直接借用关系数据库已有的成熟经验，可以和关系数据库共享信息。缺点是由于用到中间转换，将损失性能。

这种方法只是中间过渡型产品，而未来新一代数据库系统应是包括面向对象特性的，与关系数据库系统兼容的（即其语言必须是SQL的超集）成熟的数据库系统。

我们可以想象，如果新的数据库可以同时用对象和关系两种方式访问，那么对于程序员来讲就方便多了，他们可以选择自己喜欢的方式操作数据库，使得开发更加灵活。并且，企业可以将过去建立在关系数据库上的系统准确无误地移植到新的数据库系统上。而此时ORM已经不是程序员所需要考虑的了，因为它已转移到了数据库层，负责数据库系统里两种数据模型之间的相互转换。因此，只要关系存在，ORM就有其存在的价值，它不会随着OODB的发展壮大而消亡。

下转第160页 ▶▶

调 Rp1 使 VAB=0V，并调节 Rp2 为 50% 使载波端平衡。设置调制信号峰-峰值 160mV，频率 1kHZ，载波参数不变。运行仿真开关，输出波形如图 5。

从图中可以明显看出 DSB 波形的零点相位有 180 度的变化。

改变 Rp2 参数偏离 50%，适当调整示波器扫描参数会得到已调波形如图6。

从图中可以看到出现了不正常的 DSB 波形。

原因分析：Rp2 参数偏离后，使 V1，V4 输入端不平衡，这相当于在载波输入端叠加了一直流信号，出现正，负幅值不相等的输入信号，导致已调波形出现正，负（上，下）不对称的 DSB 波形。

5 结束语

在 multisim8 环境下进行调幅电路的创建，仿真非常地方便、快捷。可灵活的改变电路参数，随意观察波形地变化并研究产生原因，分析电路性能，使学生对电路的深入了解，激发学生的学习兴趣，培养学生的分析问题和解决问题的能力起到了很大的作用。

参考文献
[1]杨欣.电路设计与仿真-基于 Multisim 8与Protel 2004[J].北京:清华大学出版社发行部，2006.4
[2]张肃文.高频电子线路[M].北京:高等教育出版社，1993.

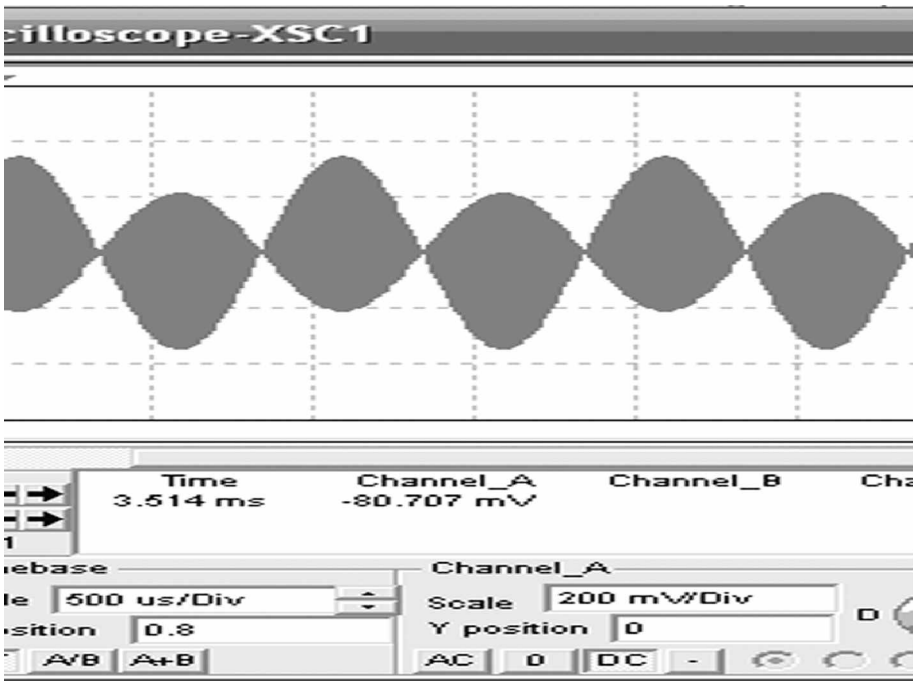

图 6 输出 DSB 失真波形

上接第 153 页

常，程序继续循环运行；否则程序转向故障处理程序，在故障处理程序中依短路电流大小进行故障判别并依短路电流大小完成电流速断保护，或者反时限过流保护，或者单相接地保护的跳闸及报警。

图 5 给出了单相接地选线程序流程图。当发生单相接地时，本系统能够判别哪一相发生单相接地和具体哪一馈线发生单相接地故障，并通过计算阻抗子子程序判断故障点的位置，便于检修。

3 结论

文中介绍了配电线路微机保护软件系统的总体结构及各个功能模块的流程图。该系统能够完成线路的单相接地短路保护和对相间短路的电流速断保护和反时限电流保护；并能够实现单相接地时线路故障点和故障类型的自动判断，便于线路的检修维护。

参考文献
[1]、陈德树.微机继电保护[M].北京:中国电力出版社,2000
[2]、张明君,弭洪涛.电力系统微机保护[M].北京:冶金工业出版社,2002

作者简介
魏臻珠（1977-），青海西宁人，讲师，主要从事电力系统分析与继电保护方面的教学与研究工作。
蒋建东（1975-），河南南阳人，讲师，博士，主要从事电力系统分析与控制方面的教学与研究工作。

上接第 156 页

参考文献
[1] 赖朝新,瞿晓静.面向对象数据库技术的两种发展途径[j].现代情报.2004，(11)
[2] 汪琛,胡浩民.面向对象数据库技术的发展与前景[j].福建电脑.2005,(5)
[3] 王意洁.面向对象的数据库技术 .电子工业出版社.2003，(3)
[4] 面向对象技术和数据库.http://www.cnblogs.com/itrust/archive/2005/01/25/97210.html
[5] O/R Mapping.研究报告 http://www.alixixi.com/xHTML_ArticlePrint.asp?classid=1&subjectid=39&articleid=14073
[6]matrix;magicgod.Hibernate特点与思考http://www.matrix.org.cn/resource/news/381_Hibernate.html
[7] 胡天平.新一代数据库技术-面向对象数据库系统.http://www.zdnet.com.cn/developer/tech/story/0,2000081602,39104104-1,00.htm
[8]面向对象数据库的现状与未来趋势.http://ced.xxjy.cn/Resource/Book/Edu/JSJCKS/TS005004/0020_ts005004.htm
[9]JDO之前世今生.http://blog.cnbie.com/blog_24056.html
[10]数据库对象的缓存策略.http://blog.csdn.net/buaawhl/archive/2004/12/21/224184.aspx