

# 微软数据访问组件应用研究

周森鑫

(安徽财经大学 信息工程学院, 安徽 蚌埠 233041)

**摘 要:**信息工程的核心系统是数据库管理系统,而数据访问技术是数据库管理系统软件开发的重点和难点。目前较为流行的开发平台是微软的 Windows 系列平台,而基于 Windows 平台的数据访问组件有很多种,根据开发环境选择合适的组件能大大提高开发效率和软件的性能。文中简要回顾微软的数据访问组件(MDAC)的发展历程,对目前常用的组件进行比较和分析并对最新的 ADO.NET 的体系结构和使用技术进行探讨。

**关键词:**微软数据访问组件;XML;ADO.NET 体系结构

**中图分类号:**TP311

**文献标识码:**A

**文章编号:**1673-629X(2007)01-0052-04

## Application Research on Microsoft Data Access Component

ZHOU Sen-xin

(Information Engineering School of Anhui University of Finance & Economics, Bengbu 233041, China)

**Abstract:** Kernel system of information engineering is database management system and data access technology is essential for development of database management system application software. Now popular development platform is serial Windows system on which there are many ways and means of data access. Selecting proper component and way can improve development efficiency and capability of application software. This paper looks back development progress of MDAC(Microsoft data access component) and make a comparison and analysis with one another. Meanwhile this paper reseaches on architecture and application technology of ADO.NET.

**Key words:** MDAC; XML; ADO.NET architecture

### 0 引言

数据访问技术是软件技术的重点之一,微软的数据访问技术的发展经历了一个漫长而曲折的过程。在 ODBC 出现之前,要实现跨 DBMS 的数据库访问是一件非常麻烦的事情。ODBC 通过统一的应用程序访问数据库的接口实现对数据库的访问,数据库内部结构对程序员是透明的。OLE DB 是一种全新的基于 COM 的技术,它采用了多层模型设计,继承了 ODBC 实现开放方式数据连接的功能,能够通过一个单一的编程接口实现对关系型、非关系型和层次结构型等数据库的访问。用传统的数据库访问技术访问基于 WEB 的数据库是一件非常困难的事情。因为网络一般都是断开的,只有在客户端提出请求时,服务器才有可能与客户端相连接。因此,必须在断开的概念下实现客户端对服务器上的数据库的访问,ADO 成功地做到了这一点,并且不需要开发者做大量的工作。与 OLE DB 相比,它具有更丰富的编程接口。ADO.NET 是 ADO 最新发展的产物。它遵循更加通用的原则,并不专

门面向数据库。它包含了所有允许数据处理的类,这些类拥有典型的数据库访问功能(如索引、排序和视图)的数据容器对象。尽管 ADO.NET 是 .NET 数据库应用程序的权威解决方案,但从总体设计上来看,它不象 ADO 模型那样以数据库为中心,而是以数据为中心。ADO.NET 是新的数据访问编程模型,为应用程序开发人员提供了一种全新的数据库访问机制,它使得数据库编程变得相当容易<sup>[1]</sup>。面对纷繁复杂的数据访问技术和版本不断更新的微软数据访问组件(MDAC)让众多编程者不知所从。

### 1 微软数据访问组件技术分析

通过微软数据访问组件(MDAC),开发人员可以连接到种类繁多的关系和非关系数据源。可以使用 ActiveX Data Objects (ADO)、开放式数据库连接(ODBC)或 OLE DB 连接到很多个不同的数据源,也可以通过由微软生成和交付或者由各种第三方开发的提供程序和驱动程序完成该操作。通过当前的 MDAC 体系结构,客户端-服务器应用程序、N 层应用程序或 Web 浏览器应用程序可以访问结构化、半结构化和旧式数据存储区。另外,通过 MDAC 这些应用程序可以使用 ADO,OLE DB 或 ODBC 灵活地访问数据。MDAC 体系结构如图 1 所示。

掌握微软数据访问组件技术的发展动态对基于微软操作平台的相关软件开发具有重要意义,哪些组件将被得

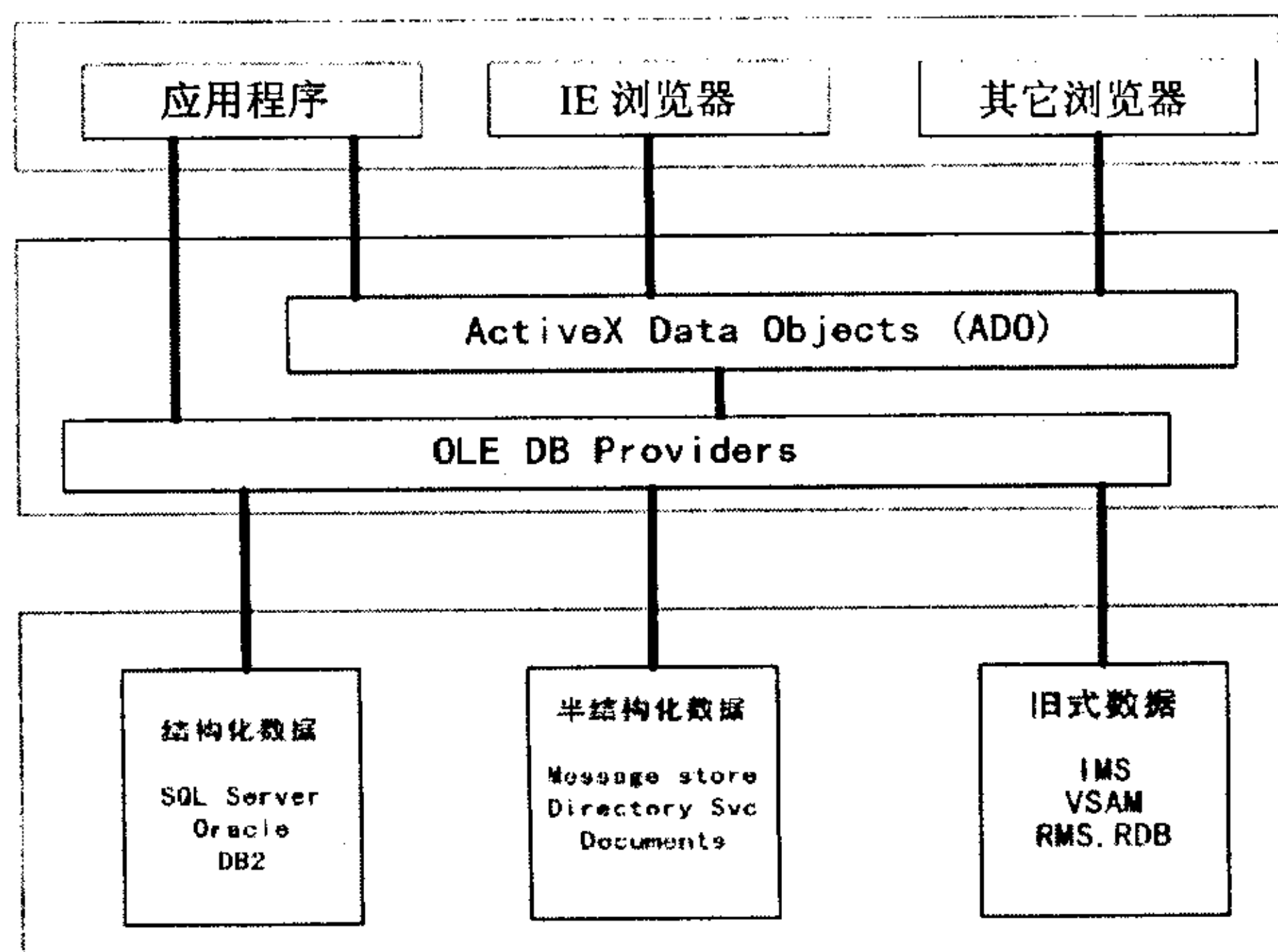
收稿日期:2006-05-04

基金项目:安徽省教育厅自然基金资助项目(2005KJ312ZC, 2005KJ049);安徽财经大学重点教研项目(ACJYD200508)

作者简介:周森鑫(1965-),男,安徽明光人,讲师,硕士,研究方向为计算机网络、计算机控制。



到加强,哪些将被弃用是软件开发人员必须关注的。根据微软发布的相关资料和信息,下列组件在以后的版本中受到支持:



**ADO:** ADO 提供了将继续得到增强的高级编程模型。尽管使用 ADO 的性能要比直接针对 OLEDB 或 ODBC 编码稍差一些,但它易于学习和使用,并且可以在诸如 Microsoft Visual Basic Scripting Edition (VBScript) 或 Microsoft JScript 之类的脚本语言中使用。

**ADOMD:** ADO Multi-Dimensional (ADOMD) 将与多维数据提供程序(例如,Microsoft OLAP 提供程序,也称为 Microsoft 分析服务提供程序)一起使用。自 MDAC 2.0 开始,没有对其进行过重大的功能增强;但是,它将可以在 64 位 Microsoft Windows 操作系统上使用。

**ADOX:** ADO Extensions for DDL and Security (ADOX) 支持对数据库、表、索引或存储过程的定义进行创建和修改。可以将 ADOX 与任何提供程序一起使用。Microsoft Jet OLE DB 提供程序为 ADOX 提供了完整支持,而 Microsoft SQL OLE DB 提供程序提供了有限的支持。在未来的 MDAC 版本中,没有为 ADOX 作进一步加强的计划;但是,它将可以在 64 位 Windows 操作系统上使用。

**OLE DB:** OLE DB 是一个全面的 COM 接口集,这些接口可用于访问多种数据存储区中的不同的数据。OLE DB 提供程序可用于访问数据库、文件系统、消息存储区、目录服务、工作流和文档存储区中的数据。OLE DB 核心服务将可以在 64 位 Windows 操作系统上使用。**SQLOLEDB:** 用于 SQL Server 的 Microsoft OLE DB 提供程序(SQLOLEDB)支持对 Microsoft SQL Server 6.5 和更高版本的访问。该 OLE DB 提供程序将是未来 MDAC 功能增强的中心,它将可以在 64 位 Windows 操作系统上使用。

**Microsoft SQL Server 网络库:** SQL Server 网络库使 SQLOLEDB 和 SQLODBC 可以与 SQL Server 数据库通信。下列 SQL Server 网络库当前在 MDAC 版本中被弃用: Banyan Vines, AppleTalk, Servernet, IPX/SPX, Giganet 和 RPC。TCP/IP、命名管道和共享内存 SQL Server 网络

库将继续得到增强,并且将可以在 64 位 Windows 操作系统上使用。

**ODBC:** Microsoft 开放式数据库连接 (ODBC) 接口是一个 C 编程语言接口,该接口使应用程序可以访问多种数据库管理系统 (DBMS) 中的数据。使用该 API 的应用程序被限制为只能访问关系数据源。ODBC 将可以在 64 位 Windows 操作系统上使用。

**SQLODBC:** Microsoft SQL Server ODBC 驱动程序 (SQLODBC) 支持对 Microsoft SQL Server 6.5 和更高版本的访问。SQLODBC 将可以在 64 位 Windows 操作系统上使用。

而另外一些数据访问组件虽然在 MDAC 的当前版本中仍然支持,但是在未来的版本中将被弃用。因此在开发软件时避免使用这些组件。这些组件列表如下:

**Jet:** 从版本 2.6 开始,MDAC 不再包含 Jet 组件。换句话说,MDAC 2.6、2.7、2.8 以及未来的所有 MDAC 版本都不包含 Microsoft Jet、Microsoft Jet OLE DB 提供程序和 ODBC 桌面数据库驱动程序。

**MSDASQL:** 用于 ODBC 的 Microsoft OLE DB 提供程序 (MSDASQL) 通过 ODBC 驱动程序提供了对数据库的 ADO 客户端访问。这已经成为 ADO 的默认提供程序;但是,对于未来版本的 MDAC 和 64 位 Windows 操作系统而言,MSDASQL 已经被弃用。因此,要从 ADO 访问数据库,客户端必须使用适当的本机 OLE DB 提供程序(例如, SQLOLEDB)来访问 Microsoft SQL Server。MSDASQL 将不能在 64 位 Windows 操作系统上使用;但是,仍然可以通过 32 位 Windows 子系统在 64 位 Windows 操作系统上使用它。

**MSDADS:** 通过 Microsoft OLE DB Provider for Data Shaping (MSDADS),可以在应用程序中创建键、字段或行集合之间的分层关系。自 MDAC 2.1 开始,没有进行过重大的功能增强。该提供程序将在未来的 MDAC 版本中被弃用。Microsoft 建议使用 XML 而不是 MSDADS。

**Oracle ODBC:** Microsoft Oracle ODBC 驱动程序 (Oracle ODBC) 提供了对 Oracle 数据库服务器的访问。它提供了对 Oracle 7 的完整支持。它还使用 Oracle 7 仿真提供了对 Oracle 8 数据库的有限支持。Oracle ODBC 驱动程序尚未针对 Oracle 9 数据库进行测试。

**RDS:** 远程数据服务 (RDS) 是一种用于跨 Internet 或 Intranet 访问远程 ADO 记录集对象的专用 Microsoft 机制。自 MDAC 2.1 开始,还没有对 RDS 进行过重大的功能增强。该组件将被弃用。Microsoft 现在交付的 Microsoft SOAP Toolkit 2.0 应用程序可以在该组件中使用开放的、基于 XML 的标准来访问远程数据。使用 RDS 的应用程序应当迁移到 SOAP。

**JRO:** 自 MDAC 2.6 开始,Microsoft Jet OLE DB 提供程序和其他相关组件已经被从 MDAC 协议栈中移除。Jet 复制对象 (JRO) 仅与 Jet (Access) 数据库一起使用,基本上



用于创建或压缩 Jet 数据库和 Jet 复制管理。JRO 已经被弃用,并且 MDAC 2.7 将是它的最后一个版本。它将不能在 64 位 Windows 操作系统上使用。

**SQL XML:** SQL XML 提供了对用于 SQL Server 的 Microsoft OLE DB 提供程序(SQLOLEDB)的扩展,以使客户端可以通过 XML 请求 Microsoft SQL Server 2000 数据以及检索 XML 流。它最初是与 MDAC 2.6 一起发布的。通过 SQL XML Web 版本 1,客户端可以使用 Updategrams 和 Bulk Load 在 SQL Server 2000 中插入、更新和删除数据。该组件将被从未来的 MDAC 版本中移除。

## 2 微软数据访问组件(MDAC)版本

随着 MDAC 的技术发展,微软推出了系列不同版本,深入了解每个版本的特点和性能对软件开发人员非常重要。MDAC 的主要版本列表如下:

**MDAC 1.5, MDAC 2.0 和 MDAC 2.1:** 这些版本的 MDAC 是通过 Microsoft Windows NT Pack, Microsoft Windows Platform SDK 或 MDAC WEB 站点发布的独立版本。它们已经被弃用。

**MDAC 2.5:** 该版本的 MDAC 随附在 Windows 2000 操作系统中。MDAC 2.5 的未来 Service Pack 将随附在相应的 Windows 2000 Service Pack 中。另外,这些 MDAC Service Pack 将根据 Windows 2000 Service Pack 发布日期安排发布到 MDAC WEB 站点中。在 Windows NT, Windows 95 和 Windows 98 平台上需重新安装该版本的 MDAC。在 Windows 2000 和 Windows Millennium Edition 平台上,只需安装相应的操作系统或它们的 Service Pack 即可使用。

**MDAC 2.6:** MDAC 2.6 RTM, SP1 和 SP2 分别随附在 Microsoft SQL Server 2000 RTM, SP1 和 SP2 中。另外,这些 MDAC Service Pack 根据 Microsoft SQL Server 2000 Service Pack 发布日期安排发布到 MDAC WEB 站点中。可以将该版本的 MDAC 和它的 Service Pack 安装到 Windows 2000, Windows Millennium Edition, Windows NT, Windows 95 和 Windows 98 平台上。

**MDAC 2.7:** 该版本的 MDAC 随附在 Microsoft Windows XP RTM 和 SP1 操作系统中。可以将该版本的 MDAC 和它的 Service Pack 安装到 Windows 2000, Windows Millennium, Windows NT 和 Windows 98 平台上。在 Windows XP 平台上,通过该操作系统或它的 Service Pack 不需重新安装。32 位版本的 MDAC 2.7 已经被发布到 MDAC WEB 站点。64 位版本的 MDAC 2.7 将只与 64 位版本的 Windows XP 一起发布。

**MDAC 2.8:** 该版本的 MDAC 随附在 Windows Server 2003 以及 Windows XP SP2 和更高版本中。在向客户发布 Windows Server 2003 的同时,32 位版本的 MDAC 2.8 也将发布到 MDAC WEB 站点中。64 位版本的 MDAC 2.8 将只与 64 位版本的 Windows Server 2003 一起发布<sup>[2~5]</sup>。

## 3 ADO.NET 访问技术分析

随着应用程序开发技术的发展,在越来越多的应用程序中使用 XML 编码通过网络连接传递数据。为了能够适应基于多层的互联网 WEB 应用程序开发,微软提出了 .NET 体系结构,ADO.NET 是其中的一个核心组件。它是为了满足新型编程模型需要而开发的,其特点是:断开式数据结构、与 XML 的紧密集成、能够组合来自多个不同数据源的数据。同时提供了对关系数据、XML 和应用程序数据的访问。ADO.NET 支持多种开发需求,包括创建由应用程序、工具、语言或浏览器使用的前端数据库客户端和中间层业务对象。其设计目标包括以下几个方面:利用 ADO 知识;支持 N 层编程模式;集成 XML 支持。传统的客户/服务器模式(C/S)主要依赖于基于连接的双层模型。当数据处理越来越多地使用多层结构时,程序员正在向断开方式转换,以便为他们的应用程序提供更佳的可缩放性。ADO.NET 编程模型是由一系列的数据库相关类和接口组成,它的基础是 XML 技术,因此通过运用 ADO.NET 技术,应用程序不仅能访问关系型数据库中的数据,而且还能访问层次化的 XML 数据。ADO.NET 为应用程序提供了两种数据访问的模式:连接模式和非连接模式。相比于传统的数据库访问模式,非连接的模式强化了应用程序的可靠性和稳定性。在该模式下,一旦应用程序从数据源中获得所需要的数据,它就断开与原数据源的连接,并将获得的数据以 XML 的形式存放在内存中。在应用程序处理完数据后,它再取得与原数据的连接并完成数据的更新工作。WEB 应用程序需要面对成千上万的用户,ADO.NET 能够有效地避免数据库阻塞和过多的活动连接占据有限的网络资源,从而在用户数量增加的情况下,系统资源的开销却增加很少。ADO.NET 有两个核心部件:数据集 DataSet 和 .NET 数据提供程序<sup>[1,6]</sup>。ADO.NET 的体系结构和两种访问模式关系如图 2 所示。

ADO.NET 中的 DataSet 类是非连接模式的核心,通过它实现独立于任何数据源的数据访问。它是以 XML 的形式存放数据。应用程序既可以从一个数据库中获取一个数据集对象,也可以从一个 XML 数据流中获取一个数据集对象。DataSet 由一个或多个 DataTable 对象的集合和有关 DataTable 对象中数据的关系信息组成。DataTable 对象由数据行(记录)、数据列(字段)以及主键、外键、约束组成。这样一个统一的编程模型可被运用于任何使用数据集对象的应用程序。从用户角度来看,数据源在哪里并不重要,也无需关心。ADO.NET 体系结构中还有一个非常重要的部分就是数据提供者对象(DataProvider),它是访问数据库的必备条件。它实现数据操作和对数据的快速、只读访问。Connection 对象提供与数据源的连接。Command 对象提供用于返回数据、修改数据、运行存储过程以及发送或检索参数信息的数据库命令。DataReader 从数据源中提供高性能的数据流。DataReader 对象是提供连接 DataSet 对象和数据源的桥梁。



DataAdapter 使用 Command 对象在数据源中执行 SQL 命令,以便将数据加载到 DataSet 中,并使对 DataSet 中数据的更改与数据源保持一致。通过它应用程序可以产生相应的数据集对象,同时它还提供了连接模式下的数据库访问支持。

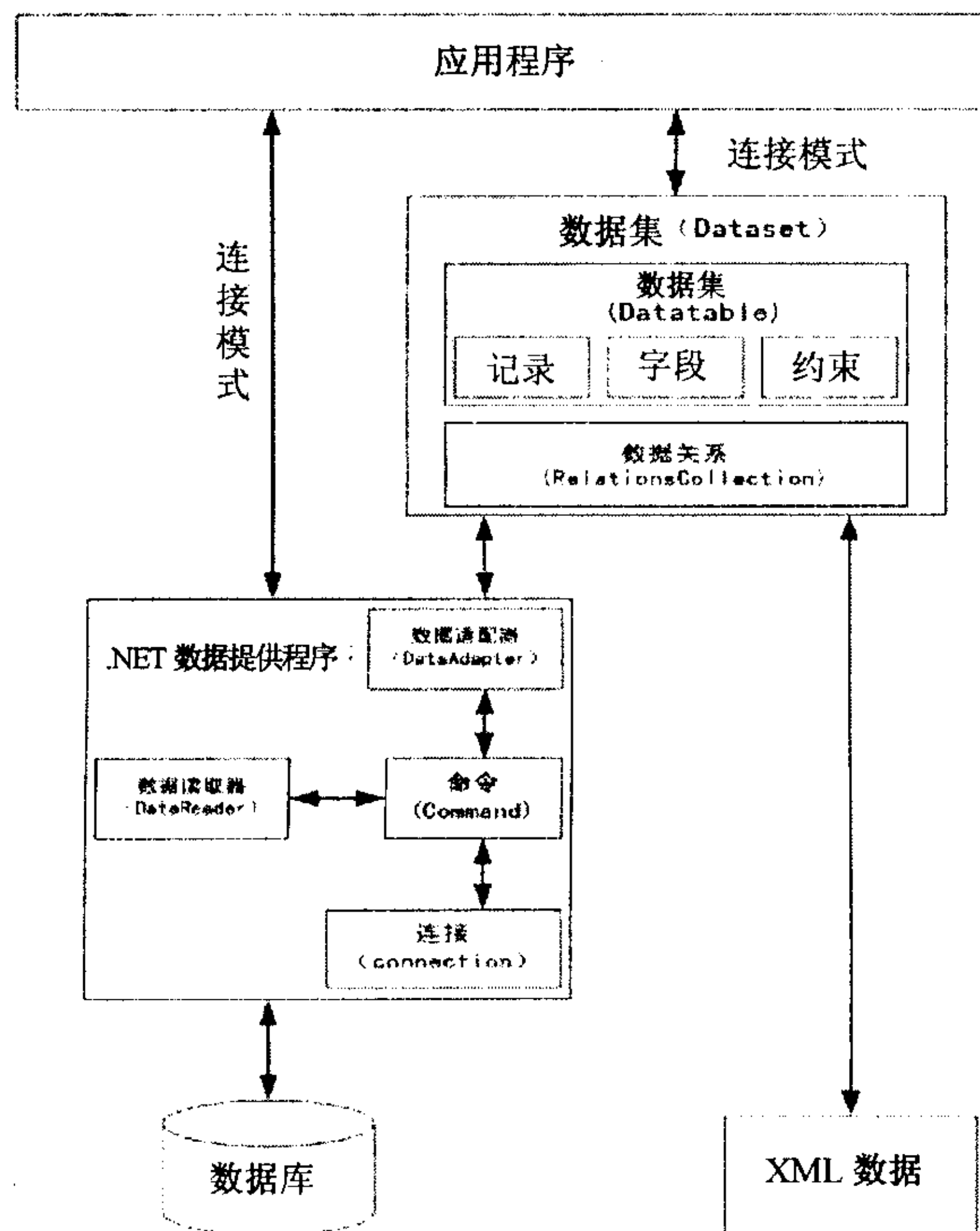


图 2 ADO.NET 体系结构图

不论从语法来看,还是从风格和设计目标来看,ADO.NET 都和 ADO 有显著的不同。在 ASP 中通过 ADO 访问数据库,一般要通过以下 4 个步骤:

- 1) 创建一个到数据库的链路,即 ADO.Connection;
- 2) 查询一个数据集,即执行 SQL,产生一个 Recordset;
- 3) 对数据集进行需要的操作;
- 4) 关闭数据链路。

在 ADO.NET 里,这些步骤有很大的变化。DataSet 是不依赖于数据库的独立数据集合。即使断开数据链路,或者关闭数据库,DataSet 依然是可用的。如果你在 ASP 里面使用过非连接记录集合(Connectionless Recordset),那么 DataSet 就是这种技术的最彻底的替代品。有了

DataSet,ADO.NET 访问数据库的步骤就相应地改变为:

- (1) 创建一个数据库链路;
- (2) 请求一个记录集合;
- (3) 把记录集合暂存到 DataSet;
- (4) 如果需要,返回第(2)步(DataSet 可以容纳多个记录集合);
- (5) 关闭数据库链路;
- (6) 在 DataSet 上做所需要的操作。

DataSet 在内部是用 XML 来描述数据的。由于 XML 是一种与平台无关、语言无关的数据描述语言,而且可以描述复杂数据关系的数据,比如父子关系的数据,所以 DataSet 实际上可以容纳具有复杂关系的数据,而且不再依赖于数据库链路。

#### 4 结束语

了解和掌握微软数据访问组件的体系结构和技术发展脉络,根据不同的开发环境选择合适的数据访问技术能大大提高软件的开发效率和软件质量。随着 .NET 的发布,基于 Windows 平台上应用程序的开发模式也发生了深刻变化,ADO.NET 在未来的数据访问技术中将成为主流技术并在 .NET 环境下占据统治地位。而微软的数据访问技术也将从面向组件技术发展为 .NET 下的面向类、接口和对象。

#### 参考文献:

- [1] 郭广军,刘永逸. ADO.NET 中数据库访问技术的研究[J]. 娄底师专学报,2004(4):62-63.
- [2] Sceppa D. ADO.NET 技术内幕[M]. 北京:清华大学出版社,2003.
- [3] 吕弘文. Dreamwaver Mx 2004 与 ASP.NET 动态网页设计[M]. 北京:机械工业出版社,2006:45-48.
- [4] Shirolka P. 数据访问技术路线图[EB/OL]. 2005-03-22. <http://www.microsoft.com/china/MSDN/library/data/dataAccess/#EFAA>.
- [5] Otey M, Otey D. ADO.NET 技术参考大全[M]. 北京:清华大学出版社,2003.
- [6] 闵娅萍. 从 ODBC 到 ADO.NET[J]. 福建电脑,2004(10):45-48.

(上接第 51 页)

- 1.]:[s. n.],1998:1765-1770.
- [7] Dasgupta D. An Artificial Immune System as a Multi-Agent Decision Support System[C]//In proceedings of the IEEE International Conference on Systems, Man, and Cybernetics[s. l.]:[s. n.],1998:3816-3820.
- [8] 莫宏伟,金鸿章. 基于免疫原理的多智能主体决策支持系统[J]. 精神疾病与精神卫生,2002,2(6):335-336.

- [9] 杨敏,邱苑华. 复杂决策系统研究——框架及其方法[J]. 系统工程理论与实践,2002(9):1-7.
- [10] Nardi B A, Miller J R, Wright D J. Collaborative, Programmable Intelligent Agents[J]. In Communications of The ACM,1998,41(3):96-104.
- [11] 叶丹,陈禹. 面向问题的动态群体决策支持系统框架研究[J]. 计算机工程与应用,2003(14):210-213.