基于.net平台与混合模式的DIMS的设计与实现①

Design and Implementation of Diabetes Information Management System Based on .Net Platform and Mixed Mode

郑炳祥 张祖平 (中南大学 信息科学与工程学院 湖南长沙 410083)
向宇飞 (中南大学湘雅附二医院 410011)

摘 要: 本文提出一种基于.net平台与混合模式的糖尿病信息管理系统的设计模型和实现方式。该系统通过数据预处理技术实现了对随访资料的有效提取;利用面向对象设计思想实现了数据库的动态创建功能,解决了数据库的灵活扩展性问题;采用开源工具NHibernate实现了数据持久层,大大降低代码编写量与出错概率,保证了系统的可移植性与健壮性。

关键词: 糖尿病信息管理系统 混合模式 动态数据库 数据预处理 持久层

1 引言

糖尿病是一种慢性疾病,医学科研机构为研究该病在不同干预治疗下的自然转归和并发症情况,建立了长期随访机制,定时随诊患者。患者长期随访资料可达10年甚至更久。由于糖尿病并发症涉及心、脑、肾、眼、足、神经、血管等多器官系统[1],临床资料丰富而复杂。如何处理和充分利用庞大的随访资料,深入挖掘可能存在的临床作用关系已成为至关重要的问题。目前一些数据库管理系统提供了糖尿病患者资料的简单录入、查询、统计功能,但都基于传统的二层或三层架构模式开发,缺乏对数据库操作的方便性和安全性。同时,由于数据库的不可变动性,使系统不适应于不同时期研究发展的需求,在一定程度上限制了系统分析管理的科学性与完整性。因此,本文提出了基于.net平台与混合模式的糖尿病信息管理系统(Diabetes Information Management System,以下简称DIMS)。该系统通过数据预处理技术使庞杂的随访资料转换成可直接利用的有效资源;在三层架构基础上增加数据持久层,实现应用程序与数据库的隔离;通过面向对象设计思想实现数据库动态更新功能。基于混合模式的DIMS系统为用户提供了方便快捷、科学准确的资料编辑、存储、统计分析和查询管理功能,为医学工作者提供了一个智能化的糖尿病信息管理平台。

2 DIMS需求与设计

2.1 系统需求分析

为能科学、方便、高效率地管理日积月累的糖尿病病案资料,DIMS需要解决的问题主要是随访资料的数据预处理、患者数据的分类管理及医学信息的统计与分析。数据预处理技术主要解决如何选取研究所需要的数据项以及如何合理规约选取出来的数据,删除不相关的属性,减少数据量。

患者数据的分类管理主要解决如何有效管理患者数据问题。患者数据包括现有患者数据与将来录入的

① 基金项目:国家自然科学基金重点项目(60433020)资助

患者数据。经多年累积，患者数据是非常庞大的，因此，数据导入到库的操作易用性是系统得以实施的重要因素。针对糖尿病类型多，数据繁杂，新类型不断增加，数据存在重复、变化等特点，系统采用面向对象的设计方法，将数据项封装成类，以可视化的组合式界面为用户提供灵活的数据项选择与创建功能，实现数据库的动态更新。

医学信息的统计与分析主要解决信息提取问题。由于信息提取条件无法确定，不同病变具有不同提取条件，即使同一病变，也会随医学发展出现不同提取条件，因此系统提供自定义提取功能，用户可以根据不同需求自定义提取条件。

2.2 系统设计

(1) 系统功能设计。该系统是集病人基本信息管理、随访资料管理、糖尿病专病资料管理、报表管理、系统管理、各类数据的分析管理等为一体的糖尿病信息管理系统。其功能模块如图1所示，包括资料录入与查询、信息统计与分析和系统管理三个子系统。

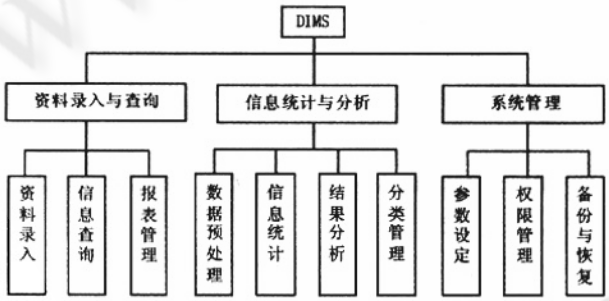

图 1　基于.net 平台与混合模式的 DIMS 的系统模块图

资料录入与查询子系统的适用对象是医护人员，主要实现各类资料的录入，并为信息统计与分析子系统提供数据基础。它由资料录入、信息查询和报表管理三大模块组成。糖尿病信息统计与分析子系统的适用对象是医生及研究人员，主要对糖尿病数据进行预处理、统计与分析操作，并进行有效管理。它由数据预处理、信息统计、结果分析和分类管理四大模块组成。系统管理子系统的适用对象是系统的管理人员，主要对该系统进行维护操作。它由系统参数设定、权限管理和数据的备份与恢复三大模块组成。

(2) 系统数据库设计。根据需求分析，系统创建两个数据库，原始数据库和动态生成数据库，其中原始数据库主要存放患者的基础信息，包括已有的数据表和由于需要而动态创建的新数据表，它为医学研究提供了全面的信息基础。动态生成数据库主要保存从原始数据库提取而生成的数据表和部分动态创建的新数据表，它主要用于医学信息的统计与分析。这种设计模式避免了直接在数据量庞杂的原始数据库上进行统计与分析，而是由结构比较简单的动态生成数据库完成，很大程度上提高了系统的响应性能，有效的解决系统效率低的问题。

(3) 运行模式设计。目前，对数据库软件的开发主要基于两种模式：客户端/服务器(C/S)模式和浏览器/服务器(B/S)模式[2]。C/S 模式结构可以充分利用两端硬件环境的优势，降低系统的通讯开销。B/S 模式是通过 WWW 浏览器实现，主要事务逻辑放在服务器端，减轻了系统维护与升级的成本[3]。系统考虑到信息的共享性与交互性，以及数据的严密性和管理性，采用 C/S 和 B/S 相结合的混合模式开发。

本系统中，内部局域网采用 C/S 模式，科室内部管理人员可通过局域网直接访问数据库服务器；而与外部网络连接时，采用 B/S 模式，用户需通过 Web 服务器才能间接访问数据库服务器。基于这种网络拓扑结构，系统的资料录入与查询模块主要利用 B/S 模式实现，信息统计与分析和系统管理模块主要利用 C/S 模块实现。这种混合模式既避免了 C/S 结构在异地查询浏览不灵活的不足，又弥补了 B/S 结构在安全性、保密性和响应速度等方面的缺陷，有效地发挥了 C/S 和 B/S 各模式的优势。

3 系统的关键技术

3.1 数据预处理技术

糖尿病病人经过多年的随访，已经积累了大量的历史数据，这些数据非常重要且不可再生，是研究疾病的危险因素和自然病程的宝贵资料。因此，对其数据的有效管理、处理、利用是一项非常必要和有意义的工作。目前，糖尿病病人的随访信息分散在患者门诊初诊登记表、随访表和1级亲属登记表等不同的文本、表格中，这些数据结果不能直接存入数据库，需对数据进行有效的预处理，转换成计算机能够操作的数据形式再存入数据库，形成直接可利用的有效数据库资源。

本系统所采用的数据预处理流程如图 2 所示，分别由数据录入、数据清理、数据变换和数据规约四部分组成，采用了多项数据预处理技术[4]。

糖尿病数据库包含许多用于研究的词库，数据录入就是将长期随访得到的数据根据各个不同研究需要存入不同的词库。但是这些随访数据不能直接存入词库，需在录入之前进行数据类型定义、数据编号转换和数据索引项建立等处理。对于表格中的连续字段，如年龄、身高等，直接采用表格中的原始数据；对于一些用选择项标识的字段，如是否高血压、是否糖尿病等数据项，就要定义一个索引项，将各个可能出现的数值用索引项表示，分别用 1、2 代表是与否。对于表格中一些多选项数据，采用各个选项分开记录的方法，将一项相应变成多项处理存入。

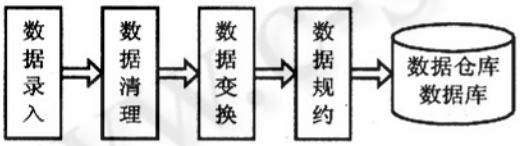

图 2　随访资料数据预处理流程

数据清理是填充空缺值、去掉数据中的噪声以及纠正数据项不一致性的过程。针对经过正确录入而存入数据库的原始数据的不完整性和不一致性，系统采用填充空缺值、纠正非法值和纠正不一致性的数据清理方法。原始数据中的一些数据项可以根据其含义定义其缺省项，而对于不能确定其缺省空缺值的数据项，如年龄、身高等数据项，可以采用全局常量去替换。对于数据的不一致性，系统通过查找数据项间存在的相关性加以纠正。对于非法值，即超过限定范围的数据，要先进行预处理转换，以免影响整体数据的处理过程和效率。

数据变换主要解决数据项中数据不统一格式的问题，以便数据进行再处理。例如，总胆固醇的单位是 mmol/L，但是有时该数据项是以 mg/dl 计量的，系统根据科学计算，设定其临界值为 20，如果数据项的值大于 20，则认为是以 mg/dl 计量，根据转换公式 f(X) =X/38 进行转换。

数据规约是通过规约算法，在保证原数据完整性的基础上，从海量数据中提取有效小数据集，提高处理效率的过程。其中，维规约技术是一种常用的数据规约算法，通过删除不相关的属性（或维）来减少数据量。属性子集的选择可以用基于子集选择的启发式方法实现。下面通过一个简单例子来说明维规约算法在本系统中的实现。假设存入数据库的原始数据如表 1 所示。其中以 dm 开头的属性集是糖尿病病史部分，以 hp 开头的属性集是高血压病史部分。

表 1　预处理前的原始数据

age	dm_a	dm_b	dm_c	hp_a	hp_b	hp_c
53	否			是	寿比山，0.1mg	120/80
56	否			否		
60	是	2001－08－10	T10/30，早 15 晚 14	否		

糖尿病病史部分的属性包括医生是否诊断过属性（dm_a）、开始使用胰岛素时间（dm_b）、目前使用的胰岛素种类和剂量（dm_c）。系统将这几个数据属性规约为一个新的属性 DM1，分别用属性值 1、2 代表患糖尿病和不患糖尿病。为了纠正数据的不一致性，规约后的新属性值不完全等同于属性值 dm，而是利用属性值 dm_a、dm_b 和 dm_c 加以修正，其实现逻辑如下：

```
if(已诊断 or dm_b or dm_c ) then DM1 = 1;
// 已患病
else DM1 = dm;//和 dm 的值一致
```

利用同样的方法可将 hp_a、hp_b 和 hp_c 规约为新的属性值 HP1。那么预处理后的数据如表 2 所示。

表 2　预处理后数据

AGE	DM1	HP1
53	2	1
56	2	2
60	1	2

表中的数据不仅保持了原始数据的各种特征，而且大大减少了存储维数，由原来的 7 维减为现在的 3 维，提高了存储效率。同时，它消除了数据间的噪声，纠正了数据的不完整性和不一致性等问题，使数据更加有效，能直接作为再处理（如数学建模、数据挖掘

等)的输入,节省了数据分析和处理的时间,当然最原始而细节的数据仍由原始数据表来保存。

3.2 数据库动态创建技术

数据库动态创建[5]是系统的重要组成部分,主要利用面向对象设计思想(OOD)实现,具有重用性、灵活性和扩展性等特点,其结构如图 3 所示。不同的糖尿病类型和研究需要不同的数据库和表,如果在整个生命周期内系统中的数据库和表都保持不变,则不利于以后的扩充与修改。因此,系统采用面向对象方法将原始数据库中已有的数据项封装成单独的类,并索引到表项感知模块。用户可以通过感知模块进行表项的灵活设计与创建,感知模块接到表项设计命令后,自动通过数据库感知控件与原始数据库进行交互,获取所需信息,并将信息传递给表项动态生成模块。表项动态生成模块主要实现表项的判定和新数据表的生成与创建功能。系统为了保证原始数据库的全面性,会自动保存用户创建的新表项到原始数据库中。

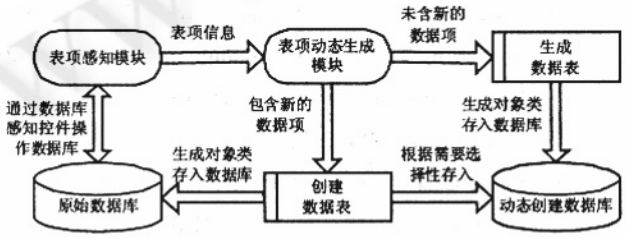

图 3　数据库动态创建结构图

数据库的动态创建是基于 ADOX[6]接口实现的。系统先引用"Microsoft? ADO Ext. 2. 8 for DDL and Security"动态链接库,再在数据库感知控件上输入数据库名称、数据表名称、字段名称、字段类型及索引名称等,通过 ADOX 接口提供的 Create 方法创建数据库,Append 方法创建数据表、字段、索引以及确定字段类型,最后释放 ADOX 的表对象和连接对象,以便实现动态表中的记录更新。

3.3 持久层的实现技术

数据持久层是系统操作数据库的一个关键系统模块,主要采用开源工具 NHibernate 实现。NHibernate 是基于. net 的 O/R Mapping 持久框架[7],为. NET 开发提供了持久化工具。它能自动产生 SQL 语句操作数据库对象,不用开发人员自己编写,从数据库底层实现持久化. Net 对象到关系型数据库的映射关系。因此,开发人员只要在确保对象提交到正确表与字段的前提下,编写代码与对象的关联即可,其开发步骤如下:

(1) 配置 NHibernate。在项目中添加了 NHibernate 后,需加入它的配置信息,在本系统中,C/S 项目的配置文件为 App. Config,而 B/S 项目的配置文件为 Web. Config。同时,也可以通过编写 MyConfiguration. cs 来对 NHibernate 进行初始化配置:

```
Configuration cfg = new Configuration( );
//定义 Configuration 对象
cfg. SetProperty ( " hibernate. connection. connection_string" ," Server = …) ; //配置数据库信息
```

(2) 创建数据库表、持久对象类和映射文件。我们通过图形化的方式或者 SQL 语句创建数据库表,而创建对象类时,应包含一些属性以及和属性相对应的 get 和 set 方法。由于所创建的表将会与应用系统中的对象类建立一一对应关系,所以在创建时要综合考虑对象类的设计。为此,系统需要在数据库表和对象类之间建立映射关系,通过配置文件实现。

(3) 持久化操作的实现。在建立持久化类和映射文件后,可以通过对对象的定义、保存、删除等实现数据的持久化操作。

```
PatientInfo ptf = new PatientInfo( );//定义 PatientInfo 对象
… //为 ptf 赋值
ISession session = factory. OpenSession( );
//打开一个新的 session
ITransaction transaction = session. BeginTransaction( );
//获取 ISessionFactory
session. Save( ptf) ; //保存对象
transaction. Commit( ) ; //执行事务
session. Close( ) ; //关闭 session
…
session. delete( ) ;//删除对象
…
```

数据持久层的实现减少了开发人员直接使用 SQL 和 ADO. NET 处理数据的时间,使应用程序与数据库产生隔离。因此,当需要将 SQL Server 数据库转换成 Oracle 或者其他数据库时,只需简单修改配置文件,而无

需从源代码级进行编写。它提高了系统的健壮性，也大大减少了程序代码量与出错机会。

4 结束语

基于.net平台与混合模式的DIMS有效避免了单模式开发的缺陷，增强了系统的安全性与灵活性。系统采用的数据预处理技术解决了数据间的噪声、不完整、不一致等问题，使数据成为可直接利用的有效资源。采用面向对象设计方法实现的数据库动态创建技术为用户提供了灵活提取与创建数据库和表的功能，满足用户不断更新的研究分析需要。同时，通过NHibernate技术实现的数据持久层降低了应用程序与数据库之间的耦合性，减少了应用程序变动对数据库造成的影响。本系统是一个具较强扩展性、灵活性和移植性的开放式糖尿病信息管理系统，它不仅满足现代化信息管理的需求，也为开发其他医学信息管理系统提供很好的参考模型。系统运行界面如图4所示。由于病人信息的保密性要求，这里只提供原始界面。

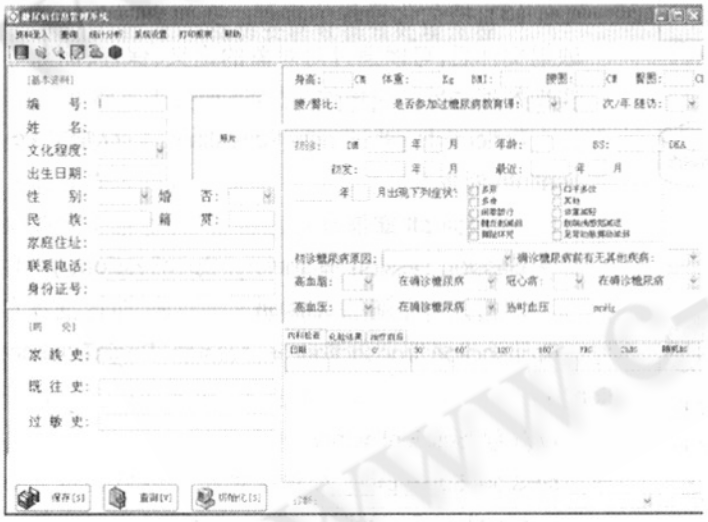

图 4　DIMS客户界面

参考文献

1 Kemal Polat, Salih Gunes. An expert system approach based on principal component analysis and adaptive neuro – fuzzy inference system to diagnosis of diabetes disease. Academic Press, Inc. 6277 Sea Harbor Drive Orlando, FL USA, 2007, 17 (4) : 702 – 710.

2 Ning Qu, Yulai Zhao. Unichos: A Full System Simulator for Thin Client Platform. Proceedings of the 2007 ACM symposium, ACM Press. 2007. 1552 – 1556.

3 彭月平、张娟子、袁涛，基于C/S和B/S结合模式的实验室信息管理系统的设计，微电子学与计算机，2006，23(8)：187 – 189.

4 Xiang Zhang, John M. Asara, et. Data pre – processing in liquid chromatography – – mass spectrometry – based prote – omics. Oxford University Press Oxford Journals Walton St. Oxford OX2 6DP UK, 2005, 21 (21) : 4054 – 4059.

5 Niraj Tolia , PM. Satyanarayanan. Consistency – preserving caching of dynamic database content. Proceedings of the 16th international conference. ACM Press, 2007, 311 – 320.

6 Pablo Castro, Sergey Melnik, Atul Adya. Fast encryption and authentication: XCBC encryption and XECB authentication modes. Matsui M, ed. FSE 2001. LNCS 2355, Berlin, Heidelberg: Springer – Verlag, 2002. 92? 108.

7 NHibernate for .NET [EB/OL]. http://www.hibernat – e.org/343.html.