工业监测平台数据服务子系统的数据库设计①

Database Design of Data Services Subsystem on Process Industry Monitoring Platform

许南山　宁　静　彭四伟　(北京化工大学 计算机科学与技术系 北京 100029)

摘　要： 为适应每个过程工业企业自身不同的企业结构和对监测不同的需求，同时也为提高软件的复用率，设计了一套通用的分布式监测平台，并详述了核心子系统-数据服务子系统的数据库结构的设计，提出了一种数据表与信息树的映射机制-将企业结构抽象成树状结构,按照树状结构设计信息表。这样可以实现自定义信息表，屏蔽数据库，实现其它子系统对服务子系统的通用查询。

关键字： 监测平台 数据服务子系统 软件复用 信息树 数据库映射

1　引言

监测系统实时监视各个设备的现场运行状况以及整个生产过程的状态，并能回溯某段时间各生产单元的运行情况，可以及时发现生产过程中的异常现象，为有关领导对调度指挥和生产管理提供决策依据，这对于企业进行安全、稳定、优化的生产具有非常重要的意义[1,2]。由于不同的生产企业有着不同的生产环境和产品，如果单独地为每个企业都开发一套监测系统则既耗时又浪费资源，软件复用率比较低[3]。因此，实时监测系统的平台化是一个良好的解决方案，它可以使企业根据自身需求选择平台提供的各种服务功能组件，自动生成一套适合于企业自身的监测系统[4,5]。此平台将极大地提高系统的二次开发效率,增强系统的可移植性，将产品进一步地商品化，组件化，同时也满足了工厂技术员的自定义过程监测系统。使方法与数据分离，成为可复用，可重构，可升级的一套软件平台。

2　结构设计思想

整个实时监测平台包括数据采集子系统、数据服务子系统、实时监测子系统(包括 B/S 模式的 web 服务子系统和 C/S 模式的数据监测子系统)和系统定制子系统。整体架构如图 1 所示：

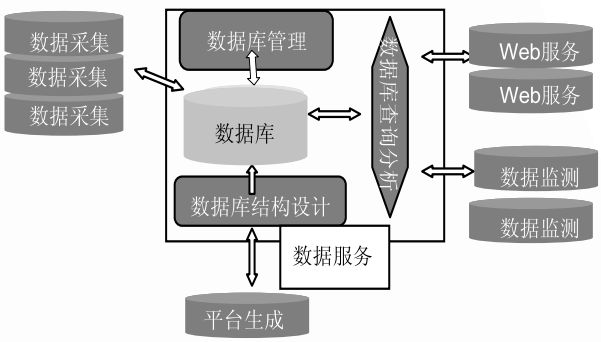

图 1　实时监测平台结构图

其中数据服务子系统是为整个平台构建核心数据服务中心，兼具数据存储、优化功能，它的设计目标是：安全性(对外屏蔽数据存储结构)、通用性(通过接口对外提供各种数据服务)、高效性(对内提高数据存储查询效率)。所包含的主要模块有：数据库结构设计模块：主要负责后台数据库的建立；数据通信接口：负责与其它子系统间数据的传输和共享；数据库管理模块：负责数据库日常的管理，包括日志的建立，数据库的维护等；数据库查询分析模块：用于优化大数据表的查询，提高访问效率等。

3　数据库结构设计

数据库结构的设计是数据服务子系统的重中之重，关键在于设计表结构以及表与表之间的关系，以用来存储各个子系统可能涉及到的数据和相关信息。

① 收稿时间:2008-12-03

从相关实际应用项目实例中可以总结出数据库存储的信息一般可以分为三类： 数据表，信息表和工程表。其中数据表用来存储通过采集程序得到的数据。信息表用来存放一些基本信息，如企业部门信息等。而工程表则是用来存放用户的工程元素信息。

由于许多企业在部门设置、工业流程管理上都有不同的方法、方式，体现在数据库中即为表结构的不同[6]。因此，像数据表一样建立统一的数据库模型很难满足监测平台灵活多变的需要，这是信息表设计时最大的困难所在。

为了克服这一难点，同时考虑到对外屏蔽数据存储结构的要求，本系统设计一种比较新颖的后台数据库结构，利用这种结构，不仅能满足上述要求，还能够实现平台内的通用查询。根据上述目标，进行信息树的构造。

一般情况下，工厂有多个车间，每个车间有多个工段，一个工段有多个流程[7]。从相关的实际应用项目来看，大多数工业企业的结构都与上述想类似，因此可以将企业结构之抽象成树状结构。因此，本子系统就是按照树状结构设计信息表。

总体设计思想：用户可以根据自己的需要定义信息树，同时，子系统将用户所建的树状结构映射成信息表。为实现树状结构，必须在每一层设置一个标志。当用户将表纳入信息树的某层时，系统自动给该表添加层的标志。

首先系统会自动生成下列三张基本表：N_info 表，T_info 表和 S_info 表。表结构分别如下：

表 1　N_info 表结构

列	数据类型	说明
结点编号	Varchar	唯一的系统标识
结点名称	Varchar	用户给结点的标识

表 2　T_info 表结构

列	数据类型	说明
信息表列名称	Varchar	唯一的列名标识，不允许重复
所属信息表	Varchar	该列名所在的信息表名
信息表深度	Int	信息表扩展结点所在的树深度

表 3　S_info 表结构

列	数据类型	说明
结点的对应列	Varchar	每层结点编号对应的列名称，如车间名称
结点深度	Int	该结点所在的深度

其中，S_info 表与 N_info 表用于描述信息树的节点结构，而 T_info 表用于描述节点的附属信息。与传统的数据库设计信息表不同，这样可以屏蔽传统的读写数据库的代码所进行的恶意访问，并且便于涉及信息表的实现通用查询，最主要还是可以适应用户自定义信息表的需要。

其次，用户可以在平台生成子系统中定义信息树的结构，系统会自动把树本身的结构数据和各节点上的数据自动映射到上述 3 个信息表中，即用户通过对信息树的自定义，来完成信息表的填充，如图 2 所示。用户首先填入对应深度的标志列名，该列名只供查询之用，系统自动将信息存入 S_info 表中；然后用户定义每个结点的值，系统将以上信息自动分配结点编号，纳入 N_info 表中；最后，用户可以自己定义表，系统将表名，表所在的层次深度和其包含的列写入 T_info 表。(写入过程:S_info -> N_info -> T_info)

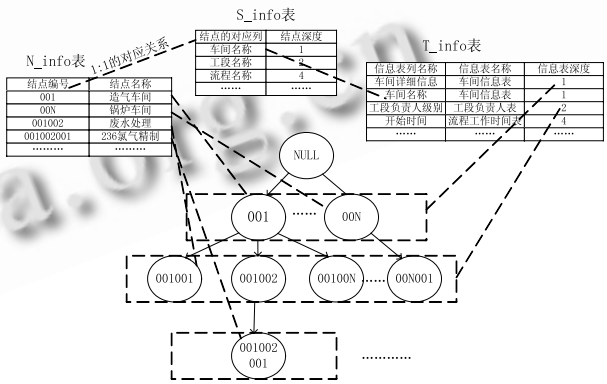

图 2　数据表与信息树的映射图

其中在分配编号时之所以加入父结点信息，是为了在整个信息表内避免命名冲突，同时能够很快建立层与层之间的纵向关系，这对实现平台化的查询有着非常重要的意义：当用户设置了每一层结点编号位数之后，通过编号的有效位数可以得到结点所在的层次。通过截取编号的指定位数可以得到它的父结点乃至祖父结点。例如，一个编号为 001045118 的结点，用户定义每层的结点数编号为 3 位，因此可以很容易得

到该结点深度为 9/3=3，其父结点编号为 001045，其祖父结点为 001[8]。

4 应用查询

下面举几个典型的例子来说明如何使用上述构造的信息树结构。

案例 1、已知某具体信息表中的某一列值，查询该列对应信息树同等深度的其它列值。

由于用户定义表名称时不同，每次查询条件和目标也不同，如何屏蔽数据库结构成为设计的关键。

例如有查询请求如下：已知车间负责人的值，想要查询该车间的详细信息。这里可能涉及到了两个表。由于在该信息树的每一层，结点值是唯一的。用户设计的每一个具体信息表都含有结点编号列。因此，本例实现如下。

步骤 1:查询系统表 T_info(含有所有的列名，所属表名，表所在树的深度)，找到含有车间负责人列的表名。

步骤 2:通过车间负责人的值查询对应的结点编号。

步骤 3:查询系统表 T_info，找到含有车间详细信息列的表名。

步骤 4:通过对应的结点编号对上表进行查询，得到车间的详细信息。

从普遍性角度，假设查询条件为 X，查询目标为 Y，在 T_info 中，找到含有 X 列名的表 X_table，并获得表深度为 hx，找到含有 Y 列名的表 Y_table，表深度为 hy。若 hx==hy，则通过以下表达式可以得到：select Y from Y_table where NoteNO in (select NoteNO from X_table where X)

函数实现：DataTable GetXByY(string X,string Xvalue,string Y)。

案例 2、已知某具体信息表中的某一个列值，查询该列对应信息树不同深度的其它信息。

上一种情况为 hx==hy，本情况假设 hx!=hy。实现如下。

步骤 1:对表 T_info 进行查询，查出 hx,hy,X_table,Y_table 的值

步骤 2:利用条件 X 对 X_table 进行查询，得到对应的 NoteNO。

若 h=hx-hy>0，即已知父结点信息，需要查询子结点信息。由于结点编号设计时，前缀包含了父结点信息，例如已知父结点编号为 004008，h=2，则需要查找的子结点为前六位为 004008，且长度为 12 位的结点编号(假定每层结点编号为 3 位)。将得到的这些结点作为条件，查询表 Y_table，即可得到 Y 值。

若 h=hx-hy<0，即已知子结点信息，需查询父结点信息。可以按照相对深度 h 的取值，截取子结点编号，得到父结点编号。然后再查询表 Y_table 得到 Y 值。

重载上述函数，实现：DataTable GetXByY(string X,string Xvalue,string Y,string Type)。

案例 3、已知信息树的深度值，求该深度上所有的结点值

该操作只需要对表 N_info 进行查询，将深度 h 与每层结点编号的位数相乘，得到该层结点的编号长度，然后将所有该长度的结点值返回。实现函数为：DataTable GetNoteByDepth(string Depth)。

5 结束语

本文介绍了数据库信息树结构以及数据接口部分方法的设计与实现，其它子系统在需要获得数据的时候，只要通过接口调用这些方法，就可以得到返回值。而这些子系统并不关心查询了哪些表，进行了哪些操作，从而达到本系统设计的目的：屏蔽数据库，方便用户自定义同时实现通用查询。

参考文献

1 杨华丽,石锐,胡捷,秦鹏.组态软件中实时数据库系统的设计和实现.实验技术与管理,2007,24(3):80－83.

2 史东林.实时监测平台数据服务子系统的设计与实现[硕士学位论文].北京:北京化工大学, 2007.

3 贺耀宜,汤利平,刘晋坤.基于 Web 的可组态实时监控系统的开发和应用.工矿自动化, 2005,2:1－3.

4 赵恒永,赵英.生产监控系统的通用化设计与实现.计算机工程与应用,1997,33(9):55－58,69.

5 林冰玉,彭四伟,汪须忠.软件开发自动化平台的研究与应用.计算机工程与应用,2005,41(9):122－125.

6 彭四伟,朱群雄.过程工业监测系统中的行为定制.控制工程, 2005,l2(4):289－291.

7 姜丽华.主动实时数据库管理系统[硕士学位论文].保定:华北电力大学, 2004.

8 赵恒永,彭四伟.过程工业监测系统中的交互协议设计.北京化工大学学报, 2000,27(4):72－74,88.