

文章编号: 1000 - 7598 - (2006) 05 - 0823 - 05

# 水工监测软件开发中数据库和数据结构设计问题

王 浩, 吴振君

(中国科学院岩土力学重点实验室, 武汉 430071)

**摘 要:** 针对水电工程监测软件系统开发, 讨论了系统总体设计、数据库和数据结构设计以及系统界面架构方面的一些共性问题, 提出用 7 种属性对象来描述水工建筑物和观测仪器之间, 以及各水工建筑物之间复杂的从属关系, 介绍了实现这些关系的数据库表和相关数据结构; 讨论了采取单文档和多窗口视界面的优点、可变大小的整编数据结构、软件容错、设置多个初值日期和更换传感器所带来的数据结构设计等问题。这些讨论对于开发监测系统软件是有益的。

**关 键 词:** 数据库; 数据结构; 从属关系; 监测; 软件系统; 水工建筑物

**中图分类号:** O 319.56

**文献标识码:** A

## Several problems of design of database and data structures for development of monitoring software system for hydraulic structures

WANG Hao, WU Zhen-jun

(Key Laboratory, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan 430071 China)

**Abstract:** Some common problems such as system architecture, interfaces, database design and data structures during developing monitoring software system for hydraulic structures are discussed. Seven property objects were proposed to describe the complicated relations between hydraulic structures and monitoring instruments and between hydraulic structures themselves. Database design and data structures are given for those objects. And several data structure problems of monitoring data process, software robust, resetting initial values and sensors replacement are discussed also.

**Key words:** database; data structures; relations; monitoring; software system; hydraulic structures

### 1 引 言

目前已经出现一些应用于岩土工程各专业领域的监测系统软件, 比如边坡、大坝、隧道和基坑监测软件等<sup>[1-3]</sup>, 这些软件各具特色。但作为现场作业人员易于操作的水电工程监测系统, 其首要目的是对监测数据进行存储、管理、整编、查询, 再提供报表制作、统计分析、曲线绘制和常规预测功能。所以其主要功能是信息管理, 核心在于一个结构良好的数据库和高效的数据结构, 能够有效地描述水电工程监测中存在的非常复杂的被监测对象和监测对象之间的从属关系。现有的监测软件对这方面缺乏系统的设计, 所考虑的对象一般只有项目、断面、测点<sup>[3]</sup>, 因此, 不能准确地描述复杂的从属关系, 导致查询不够灵活, 系统伸缩性较差。笔者

在开发某地下厂房施工期监测信息管理、预测预报系统的过程中, 就有关水电工程监测软件系统开发中的系统总体设计、数据库和数据结构设计以及系统界面架构方面总结出一些共性问题, 讨论如下。

### 2 系统总体设计

在软件编制过程中, 系统的总体设计非常关键, 它关系到项目顺利进展的程度和是否能够达到预期的开发目标。监测软件的开发同样一定要符合一般软件的开发流程。首先, 需要认真做好用户需求分析、功能设计、菜单设计、数据库设计和数据结构设计。特别是在多人合作开发的情况下, 有关文档(如大到系统框架, 小到菜单、函数模块设计)越详细越好, 便于分工和协调。这方面可以参照以下 3 个规范<sup>[4-6]</sup>:《计算机软件开发规范》(GB8566-

收稿日期: 2005-03-25

修改稿收到日期: 2005-07-01

作者简介: 王浩, 男, 1972 年生, 博士研究生, 助理研究员, 主要从事岩土工程监测、反馈分析和相关软件开发方面的工作。E-mail: wanghao325@263.net。

88) 《计算机软件产品开发文件编制指南》(GB8567-88) 《计算机软件需求说明编制指南》(GB9385-88)。其中特别重要的是用户需求分析,只有在认真分析用户需求的前提下,才能做好系统总体设计和系统功能设计。

### 3 数据库和数据结构设计

#### 3.1 从属关系的描述

监测系统所管理的最基本的对象是测点,它既可以视为内观监测的单支仪器,又可以作为外观监测的观测点。在数据库中,首先需要设计测点的属性。比如,对于一个监测点,用户可能需要了解它的以下属性:所属仪器类型、仪器参数和仪埋档案,安装于哪个钻孔,坐标位置,所属监控断面,所属水工建筑物,所属标段等等。可见这些属性既包含测点本身的基本性质,也反映了水工建筑物之间嵌套的多层从属关系。

已有的监测系统一般是采用项目、测点或者断面、测线、测点分类方法来识别监测项目中测点的从属关系,实际上,这种简化处理难以有效地描述和管理水工监测中建筑物和仪器之间的十分复杂的从属关系,直接导致后续的查询不够灵活,查询功能不够强大。

通过分析水电工程各种监测信息之间的逻辑关系,笔者概括出以下 7 种对象作为系统数据库组织的基本单位,即:工程项目、标段、单元工程、断面、组、测点类型、测点。

工程项目代表某个具体的监测项目,项目可以包含多个标段。每个标段具有独立的施工承包单位,标段中可以包含多个单元工程。单元工程被视为一个标段内同时施工,具有掌子面(工作面)或其他可以统计施工进度的分部工程,比如地下厂房中的交通洞、主厂房等水工建筑物。每个单元工程可以拥有多个监测断面。每个断面可以拥有多个测点(仪器)类型;每个测点类型下面可以拥有多个组,也可以没有组而直接拥有测点。笔者定义的组的概念

比较广泛,主要是将某些在空间或者时间上具有共性的测点集合视为一个组,比如位于一个钻孔中总是被同时观测的多点位位移测点集合构成一个组;或者对于同一个断面下的收敛计测线,按不同的施工工序,如中导洞开挖、扩挖以及分层开挖时,将它们视为不同的组。测点可以是内观埋设的单支仪器,也可以是一条收敛计测线或者外观变形监测点。

显然,通过以上 7 种对象就能够非常有效地建立水工建筑物和观测仪器之间,以及各水工建筑物之间复杂的从属关系。同时,可以看出,这种归纳既考虑了空间从属关系,也顾及到时间上的施工顺序。另外,我们还定义了掌子面对象,掌子面对象属于单元工程,一个单元工程可以拥有多个掌子面。系统可以分别跟踪各个掌子面的施工进度,进行有关监测日志的建档及施工日志、施工进度的动态管理。

采用以上 7 个对象加上掌子面对象(7+1),既能管理非常复杂的水工监测,又能管理仅仅一条隧道的监测信息。这样设计的监测系统软件伸缩性好,适应性很广,并且数据库冗余小,便于实现统计查询。

#### 3.2 从属关系的数据库实现<sup>[7]</sup>

在关系数据库中,数据保存在表中,各个表通过主键和外键的关联建立关系。分别设计上述各属性对象的数据库表,就能够储存属性数据和描述它们之间的从属关系,为下一步的查询创造条件。下面以 Microsoft SQL Server 2000 为例说明它们在数据库中的实现。

一个项目对应着一个数据源和数据库,所有的信息都保存于一个数据库文件中,既显得简洁,又可以防止用户对文件的误操作影响系统的运行。系统能管理多个项目,但系统的一个应用实例同时只能打开一个项目。其他 6 个表(标段、单元工程、断面、组、测点类型、测点)及其之间的关联如图 1(限于篇幅,每个表只介绍主要字段)。

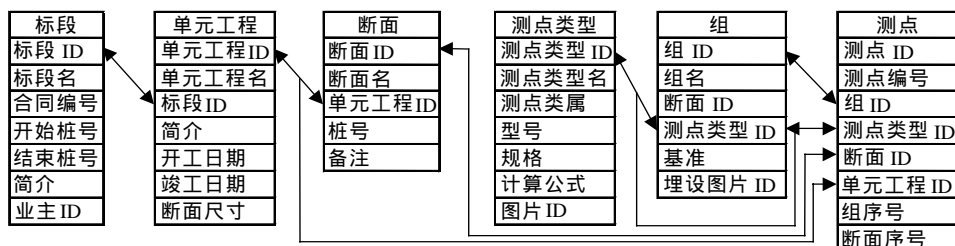


图 1 6 个属性数据库表及其关联

Fig. 1 Database table of 6 property objects

数据库中“属性对象表”列示出“对象索引、对象名、对象类型、对象 ID、父索引、测点类型”，记录了系统中的所有属性对象，并通过“父、子索引”表达了它们之间从属关系。“父索引”就是父对象的“对象索引”。

### 3.3 从属关系的编程实现<sup>[8]</sup>

下面以 Visual C++ 为例说明它们在应用程序中的实现。Visual C++ 具有非常强大的类型安全的数据结构，笔者在软件开发中广泛使用了映射类 CMap、动态链表类 CList、动态数组类 CArray 为基类来派生自己的数据类。

在界面上采用层次型树控件 (CTreeCtrl) 来表达属性对象表，树中的每个节点代表一个属性对象，树节点的 ItemData 设置为属性对象的“对象索引”，树节点的层次以及父子关系就代表了各个属性对象之间的从属关系。

树节点结构 stTreeNode 的成员为数据库属性对象表中的字段，类 CMapIndex2Node 将属性对象的“对象索引”映射到树节点结构。

```

struct stTreeNode
{
    TCHAR chName[MAX_NUM]; //属性对象名称
    int nType; //属性对象类型值,即7+1种类型
    HTREEITEM hItem; //节点的树句柄
    int lParentIndex; //父节点的索引
    int nDBID; //节点在数据库中的ID
    int nPnType; //节点测点类型
    .....
};

class CMapIndex2Node: public CMap
<int,int&,stTreeNode*,stTreeNode*&>
{
public:

```

```

CMapIndex2Node();
void Serialize(CArchive& ar);
~ CMapIndex2Node();
.....
} m_MapIndex2Node;

```

系统初始化时,首先从数据库中读属性对象表得到所有属性对象,然后按照父子先后顺序生成树控件节点,并填充变量 m\_MapIndex2Node, 树节点的索引被赋值为属性对象表中的“对象索引”,并被映射到树节点结构。

每新增一个属性对象时,系统生成一个节点结构加入 m\_MapIndex2Node 中,并在数据库属性对象表中添加一个记录;删除属性对象时,同时从 m\_MapIndex2Node 中删除节点结构和从属性对象表中删除对应记录。很明显,数据库属性对象表的记录是按父子先后顺序排列的。

这样,通过树节点的 ItemData 得到属性对象的“对象索引”,通过 m\_MapIndex2Node 得到树节点结构,通过节点结构可以知道该节点的类型及其在数据库中的 ID、测点类型等,进而通过 ID 查询到各个属性数据库表,就掌握了整个系统中的属性对象和它们之间的从属关系。其关系如图 2 所示。

### 3.4 可变大小的整编数据结构实现

系统中测点对象关联着监测数据和仪器参数表,为每类仪器设计了一个测点数据表。有的仪器(如多点位移计)总是按整个组或者整个断面的测点一起存储、输入、整编和显示过程曲线,笔者采用组 ID/断面 ID 来识别同一个组和同一个断面中的测点。多点位移计的数据库记录表格格式如下:“组 ID、观测日期、测值 1~测值 6、位移 1~位移 6”。

由于一个组中测点的个数不确定,可采用结构 stZBData 和动态数组 CZBDataArray 来解决数据整编问题。

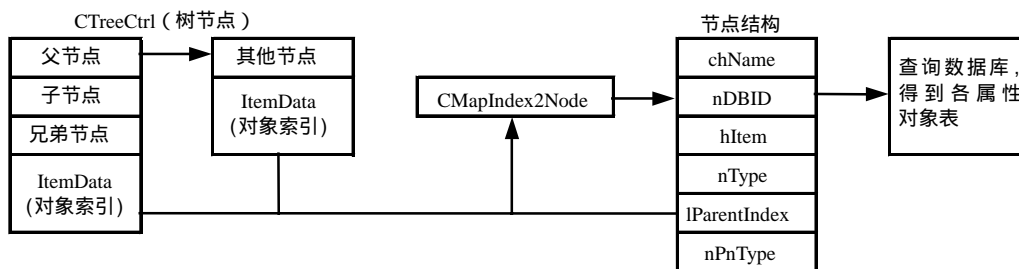


图2 树节点-节点结构-数据库表之间关系  
Fig.2 Relationship between treenode, nodestructure and database table

```

struct stZBData
{
public:
    stZBData();
    stZBData(const stZBData& src);
    stZBData& stZBData::operator =(const stZBData
&src);
    CMyFloatArray aVal;//测值，单精度数值数组
    CMyDoubleArray aSum;//累积位移，双精度数
值数组
    double fDate;//日期
    .....
};
class CZBDataArray: public
CArray<stZBData,stZBData&>
{
public:
    CZBDataArray();
    void CopyZBData(const CZBDataArray& src);
    .....
};

```

其中 aVal 和 aSum 为动态数组，维数不定。比如 m\_aZBData 为 CZBDataArray 变量，对于某组数据，第 i d 的原始记录存于 m\_aZBData[i].aVal，累计位移存于 m\_aZBData[i].aSum，如果组内有 5 个测点，则 aVal 和 aSum 的维数为 5。这样在编程上就比较简洁、统一地实现在测点个数未知情况下整个组/断面测点原始数据的存储。

### 3.5 单文档和多窗口视 (Form View) 界面的实现

一般监测系统采取数据表格，对话框或单 FormView 三种数据录入界面。这些界面其实都存在不直观，操作不便和不能同时打开、查看多个相关窗口等缺点。为此，设计了单文档和多 FormView 结构，具有多文档界面的风格，可以同时打开多个输入的 FormView 并随时切换，便于数据录入和查看。比如，用户在输入某测点监测数据时，能够同时看到该测点的属性窗口。其编程思想如下：

- (1) 基于对话框资源创建一个 FormView 的 C++对象 pView；
- (2) 用系统缺省的文档模板对象创建一个子框架类对象 pFrame；
- (3) 为子框架对象创建并添加一个 pView 的 Windows 窗口对象，销毁第二步创建 pFrame 时缺

省生成的 View 对象，最终形成一个子框架类对象包含一个指定类型的 FormView。

程序片断参考如下：

```

CFVBidInfo*pView=new CFVBidInfo(nIDTemplate);
// CFVBidInfo 是标段属性 FormView，nIDTemplate
为对话框资源 ID，并且有对应的对话框模板
...
pFrame=(CChildFrame*)(theApp.GetGeneralDocTem
plate()->CreateNewFrame(this,NULL);
pView->Create(NULL,NULL,style.rc,pFrame,nIDTe
mplate);
AddView(pView);
pFrame->SetActiveView(pView);
.....

```

### 3.6 表格控件的容错问题

监测系统软件需要具有很强的容错功能，对用户的误操作、误输入会自动提示和取消输入。Visual C++ 对常规控件提供了对话框数据验证功能 (DDV)，但对于输入中常用的表格控件 (MSFlexGrid) 没有提供相关功能。为此，设计了结构 stCellCtrl 和类 CMapCol2Ctrl 来实现表格控件的容错。

```

struct stCellCtrl
{
    EnumCellControlType nCtrlType;//指示单元格
控件类型
    EnumCellTextType nCTT;//单元格文本类型
    TCHAR chFormat[18];//单元格文本格式
    double dMin,dMax;//取值范围
    COleDateTime dtMin,dtMax;//取值范围
    UINT nMaxChar;//字符串最大长度
    CStringList strList;//如果单元格要显示
ComboBox，则strList记录将要填入ComboBox的内
容
    bool bCanNotBeNULL;//本列是否允许空
    .....
};
typedef CMap <long,long&,stCellCtrl*,stCellCtrl*&>
CMapCol2Ctrl;

```

通过 CMapCol2Ctrl 将 MSFlexGrid 的每列映射到结构 stCellCtrl，当双击 MSFlexGrid 的单元格时可以根据结构成员 nCtrlType 的取值显示编辑框、下拉列表 (ComboBox)、图片浏览器等任何其他控件，

同时，通过stCellCtrl结构的几个取值范围成员实现输入数据的验证。

另外，还可以在数据库中设置约束等来防止输入出错，但为了减轻服务器的压力，最好直接在客户端软件中进行数据验证。

### 3.7 多个初值日期的考虑

监测实施过程中，测点被破坏导致需要重设初值日期或者重新更换传感器的现象时有发生，监测系统软件必须能够处理这种情况。为此，设计了测点变动表将变动测点的ID、发生变化时的日期以及其他信息记录于数据库中，并用类CIniDateList记录某组/断面中所有测点的全部初值日期。在数据整编过程中系统会扫描CIniDateList，如果发现该天是初值日期，再做相应处理。

```
struct stIniDate
{
    int nPnID;//测点在数据库中的ID
    double dbDate;//日期值
public:
    stIniDate& stIniDate::operator =(const stIniDate& src);
    .....
};
class CIniDateList: public
CList<stIniDate*,stIniDate*>
{
public:
    bool FindContent(const stIniDate&
st,POSITION& posFind);
    void GetFirstIniDate(const int nPnID,double&
dbIniDate);
    void CopyList(const CIniDateList& src);
    bool IsIniDate(double dbDate);//检查指定的日期是否是初值日期
    .....
};
```

对于多次更换传感器的情形也可以照此处理。

## 4 结 论

本文探讨了水电工程监测软件系统开发中的系统总体设计、数据库和数据结构设计以及系统界

面架构方面的几个共性问题。

(1) 提出的7种属性对象能非常有效地建立水工建筑物和观测仪器之间，以及各水工建筑物之间复杂的从属关系，并提出实现这种关系的数据库表和相关数据结构，按这种思路开发的系统伸缩性好；

(2) 讨论了采取单文档和多窗口视的系统界面的优点；

(3) 探讨了可变大小的整编数据结构、软件容错、设置多个初值日期和更换传感器所带来的数据结构设计问题。

笔者认为这些讨论对于水工以及岩土工程监测软件系统的开发人员是相当有益的。

## 参 考 文 献

- [1] 李元海, 朱合华. 岩土工程施工监测信息系统初探[J]. 岩土力学, 2002, 23(1): 103 - 106.  
LI Yuan-hai, ZHU He-hua. Development of monitoring information system software for geotechnical engineering[J]. **Rock and Soil Mechanics**, 2002, 23(1): 103 - 106.
- [2] 王浩, 葛修润, 邓建辉, 等. 隧道施工期监测信息管理系统的研制[J]. 岩石力学与工程学报, 2001, 20(增): 1 684 - 1 686.  
WANG Hao, GE Xiu-run, DENG Jian-hui, et al. Study on the management system of monitoring data in tunnel excavation[J]. **Chinese Journal of Rock Mechanics and Engineering**, 2001, 20(Supp.): 1 684 - 1 686.
- [3] 孙钧, 胡向东. 盾构隧道施工监控系统数据库的研究[J]. 地下工程与隧道, 2002, (1): 2 - 9.
- [4] GB8566 - 88, 计算机软件开发规范[S].
- [5] GB8567 - 88, 计算机软件产品开发文件编制指南[S].
- [6] GB9385 - 88, 计算机软件需求说明编制指南[S].
- [7] Microsoft Cooperation, Microsoft SQL Server 2000 联机帮助[Z].
- [8] Microsoft Cooperation, MSDN Library 2000, Microsoft Visual C++ 联机帮助[Z].