

水文信息通用化管理模式研究

王船海 虞玉诚

(河海大学水资源环境学院 ,江苏 南京 210098)

摘要 通过讨论传统水文信息管理系统中存在的不足 ,分析水文信息的特点 ,采取面向对象的思想 ,通过站网及测站对象的动态管理、数据库的动态连接器的设置 ,实现了水文信息通用化管理。
关键词 水文信息通用化管理 ;面向对象编程 ;中间件 ;VC++
中图分类号 :P338 **文献标识码** :A **文章编号** :1000-1980(2005)06-0620-04

水文信息在水文系统的水文软件开发中起着关键的作用^[1]。水文信息是应用程序开发的基础 ,如何对各种格式甚至异种的信息库中的信息进行有效的利用 ,一直是水文软件开发者所关注的热点问题之一。本文对水文信息的管理提出了一种新的思路 ,突破了以往确定型信息库表结构的水文软件开发模式 ,对水文信息采取了一种通用化的管理方式 ,使得所开发的软件能够适用于各种水文信息库系统。

1 水文信息管理的现状

1.1 水文信息库的特点

现有水文信息库的特点就是存在着多种库表结构、多种信息库系统平台的特点^[2]。水文软件中对水文信息进行处理一般是针对于某种库表结构进行开发。这样的处理方法在一定程度上优化了软件 ,针对性较强 ,但在水文信息通用化的系统维护以及可扩充性等方面存在着以下的不足之处 ,不利于水文信息管理系统的推广和应用。

1.2 水文信息管理中存在的问题

- a. 系统可维护性差。信息库中某些信息格式或处理公式发生变化 ,客户端程序便得大幅修改 ,接着重新编译、连接、生成可执行程序 ,然后再安装到所有使用者的计算机上。这不仅会使项目的应用程序因为信息库管理复杂而导致开发周期的延长和开发成本的急剧上升 ,而且对客户端进行管理和维护相当烦琐 ,这在小规模的 LAN 中尚能实现 ,但在随着网络化的进程的加快和信息密度的增加 ,这将变得非常困难。
- b. 可扩充性差。目前水文软件针对水文项目的各个信息项比较固定 ,软件开发相对容易。随着测站设施的逐步完善 ,一个水文测站如果遇到测站信息项的增添 ,那么所开发的软件将很难继续适用。应用软件为了满足用户不断发展多样化的需求 ,客户端所安装的应用程序的规模也与日俱增。
- c. 局限性强。目前水文信息存在着多种的库表结构 ,信息库表结构不统一 ,使得所开发的水文软件存在着很大的局限性。现有的水文软件一般是针对某个特定的信息库进行开发 ,为一个项目开发的软件难以应用于另外的地区 ,软件的重复开发 ,在资源的利用上造成了很大的浪费。

2 问题的分析和解决

针对水文信息管理中存在的不足 ,在程序开发中采取了通用化信息管理的思想。将应用程序分解为水文信息应用层(下面简称为应用层)与信息获取和处理的获取层(下面简称为获取层)2 部分。应用层通过内部接口调用获取层标准化后的信息 ,对于获取层如何对获取信息不用关心。获取层的主要的功能就是实现和外部信息的交互 ,将外部信息转换成一个统一的信息格式 ,然后通过接口被应用层所调用 ,见图 1。

信息获取层划分为如下 3 个层次 :第 1 层次是用于站网管理 ,第 2 层次是用于配置相应站网类别与数据

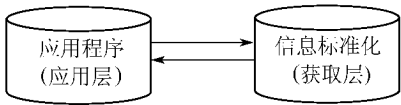


图 1 应用程序结构

Fig.1 Structure of application program

库连接属性,这 2 个层次上的对象是一一对应的,第 3 层次是用于管理具体水文测站信息,具体结构见图 2。

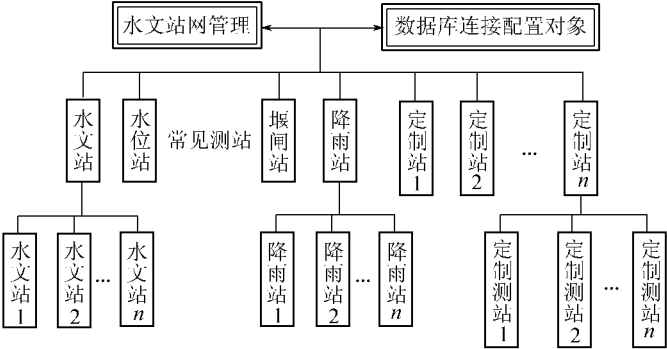


图 2 信息获取层结构

Fig.2 Structure for information acquisition

水文站网管理通过类 CHydroDataItem Type 实现,该类管理着水文站网类别的增减等功能,解决测站类别的扩充性问题,如:添加水环境测站类,其类结构表述如下:

```
class CHydroDataItem Type : public CObject
{
    .....
public :
    int m_KeyID , //惟一标识测站类别的 ID 值
    CString m_Name , //站网类别的名称,如:水文站、水位站、堰闸站等
    unsigned char( * m_HydroInfo ][ 20 ], //对测站子项的说明,如水位、流量、闸上水位等
    int m_DataNumber , * m_Type , //数据子项的个数及子项的数据类型,如系统内部针对
    //不同量纲信息用不同数值标志,便于应用层中的应用
    CDb sSqlInfo m_DbsSqlInfo , //存放数据库中连接配置对象
    .....
}
```

数据库连接配置对象主要是解决相应的站网类别与数据库相连的配置信息问题,将各个测站规定为不同类型的依据就在于测站所测量信息项的不同,但其又具有一定的相似性。水位站的主要信息要素有:站码、时间、水位;堰闸站的主要信息要素有:站码、时间、闸上水位、闸下水位;雨量站的主要信息要素有:站码、时间、降雨量、降雨历时;蒸发站的主要信息要素有:站码、时间、蒸发量、时段间隔,还有一些测站信息就不一一列出了。通过分析可以发现,可以看出这些测站的信息可以分成 2 类,一类为瞬时信息,另一类是时段信息,像水位站、堰闸站的信息就是瞬时信息,而雨量站、蒸发站的信息则是时段信息。站网管理和数据库连接属性这一层次的具体数据结构表述如下:

```
class CDb sSqlInfo : public CObject
{
public :
    CString m_DataTable , //测站信息在数据库中的表名
    m_DataTime , //测站信息的时间项
    m_DataCode , //测站信息所对应的站码
    m_DataSQL , //测站信息所对应的 SQL 项
    m_StationCodeSQL , //测站站码所对应的 SQL 项
    m_DataWhere , //条件语句
    LONG m_CodeType , //确定站码的类型
    .....
}
```

站网管理和数据库连接配置对象之间是一个一一对应的关系,数据库连接配置对象解决了某一站网类

别是从数据库什么表以及那些字段中获取相关信息的问题.形象地说就像一个索引,引导站网管理类找到目标信息,其数据结构如下:

```
class CHydroDataItem : public CObject           //测站具体信息管理,与 CHydroDataItemType 结合使用
{
    .....
public :
    int m_KeyID ,                               //惟一标识具体测站的 ID 值
    CString m_Name ,
    CHydroDataItemType * m_HydroDataItemType , //指向 CHydroDataItemType 的指针
    char mc_HydroStationCode[ 20 ] ,           //与数据库相对应的水文测站站码
    .....
public :
    virtual BOOL GetCurData( double * m_AO ,time_t CurTime ,LONG TimeSpace ) ,
    //获取瞬时值接口 ,TimeSpace 是一个时段长变量 ,单位为秒 ,该值对时段信息起作用 ,
    //对瞬时信息不起作用 ,如对雨量站起作用
    virtual BOOL GetAverageData( double * m_AO ,time_t CurTime ,LONG TimeSpace );
    //获取平均值接口 ,该值对时段信息不起作用 ,对瞬时值起作用
    .....
}
```

信息处理的过程,也就是上面所说的第3个层次,是信息通用化管理关键的一部分.根据上面对水文信息的分析可以看出,不同的信息存在不同的特点,一般可以分为时段信息和瞬时信息这两种.各类测站通过图2所示的测站类对具体测站进行信息获取,然后再通过一系列接口由站网管理类向数据库提交配置信息,从而向数据库提出要何种信息的请求,根据这个请求数据库按照一定的格式返回信息,如果得到不是所需要的信息可以进一步处理,最终得到标准化了的信息.在上述数据结构中列出了两个信息获取的接口,在实际应用中根据不同的情况灵活地设置接口以解决相应的问题.

通过严谨的层次结构和灵活的内部接口方式将信息通用化管理思路得到了具体的实现.一系列的具体处理过程都被封装在一起,只是将处理好的信息按照要求提交给应用层,而这个响应的过程如图1所示.这样的通用化处理方式解决了水文信息通用化的系统维护以及可扩展性等方面存在的不足,具体优点如下:

a. 应用层只需要处理固定格式的信息,而不用理会获取层是从何处或何种方式获取信息,使得系统维护简单、方便.

b. 设置了各种接口用以处理不同的情况,统一规划,统一接口标准,使得子系统间容易集成,还可以方便地增添不同的接口解决测站的增减或者其他会产生的新问题,容易扩展、升级.

c. 对信息进行了有效的处理,使得信息使用能够通用化,打破了水文软件存在的局限性,不必因为信息源的不同而进行重新的开发,优化了资源.

d. 根据应用层的实际需要结合测站的具体特点索取信息,不会产生信息冗余的问题,使得信息的使用高效化、精确化.

3 水文信息通用化管理的进一步讨论

由于采取面向对象的程序设计思想^[3],在程序设计中实现了逻辑层次的概念,以内部接口的方式实现具体的信息统一管理,按照这种思路可以灵活地将应用软件设计成2层或3层模式.

2层结构是将应用程序和信息处理封装起来,使用户不需要考虑内部的信息结构和信息如何被调用,操作简单、易上手,只要按照相应的操作步骤就可以完成信息的调用和管理.针对一个信息库进行各信息项配置之后,这个配置方案作为一个配置信息保存下来,若干个信息库就可以相应的有若干个配置信息,而且对于不同得的信息类型也可以生成相应的配置信息(图3).以后再遇到该信息库时,只要导入相应的配置信息就可以完成应用程序和信息库之间的联系,而不用再次重新配置,使得应用程序能够在不改动底层应用程序代码的基础下,达到各种不同信息类型为我所用的目的.

3 层结构则可以引入目前所流行的中间件概念^[4],将信息处理模块作为一个中间件(Middleware)独立出来,以满足不同用户的需求.3 层结构的信息管理和访问方式更具灵活性,合理地分割 3 层结构并使其独立,可以使系统的结构变得简单清晰,提高了程序的可维护性和可扩充性.除了满足通用化管理的目的之外,它具有 2 个明显的优点:

- a. 客户端向在第一次连接信息库时,通过中间件对信息库的访问和连接进行一系列地配置,并且这些配置信息作为一个文件存放,由于文件中只是一些配置信息,因此其容量就可以很小,不会占用大量的空间.
- b. 可以解决多操作系统的问题.根据操作系统不同生成客户端的应用程序后,只需利用中间件生成相应的配置文件,就可以实现多种系统下通用化信息管理.应用程序生成之后就不需要因为信息管理方面的问题而进行客户端程序的大幅修改.

4 结 束 语

水文信息通用化管理模式可以降低整个系统的成本,不仅使系统的维护升级十分方便,而且具有良好的开放性和可扩充性,还可以进行严密的安全管理,可支持异种信息库,具有较高的实用性.该处理模式已广泛应用于数字流域系统中,如三峡区间入库洪水实时预报系统^[5]、太湖流域水资源综合规划、湖南省资水流域洪水预报系统等应用系统,应用效果较好.

参考文献:

[1] 莫渭浓,张建云.国家防汛指挥系统概述[J].水文,1998(增刊) 36—40.
[2] 张建云,舒云卿,李键.水文情报预报系统开发中的若干问题探讨[J].河海大学学报(自然科学版) 2000,28(1) 86—90.
[3] 王育坚. Visual C++ 面向对象编程教程[M].北京:清华大学出版社,2003.49—52.
[4] 张安勤,周晓峰.基于构件技术的中间件研究[J].安徽师范大学学报(自然科学版) 2004,27(4) 391—394.
[5] 王船海,郭隶君,芮孝芳,等.三峡区间入库洪水实时预报系统研究[J].水科学进展,2003,14(6) 677—681.

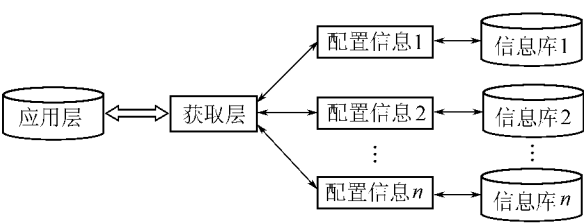


图 3 多方案信息管理

Fig.3 Multi-scheme information management

Universal hydrological information management model

WANG Chuan-hai , YU Yu-cheng

(College of Water Resources and Environment , Hohai University , Nanjing 210098 , China)

Abstract: The disadvantages in the conventional hydrological information management were analyzed. Based on the characteristics of hydrological information , the idea of object-oriented programming(OOP) was suggested to overcome the disadvantages and realize the universal hydrological information management. The universal management was realized by strict hierarchical structure and flexible interface linkage , and the management mode is of good openness and expansibility. With low cost in system development , the system is convenient in mode maintenance and update , and strict in safety management. It can support different kinds of information databases. Therefore , it is practicable in hydrological software development.

Key words: universal hydrological information management ; object-oriented programming ; middleware ; VC++