

文章编号: 1005-0523(2000)04-0050-05

SCADA 系统中数据库子系统的面向对象建模

杨 丰 萍

(华东交通大学 电气信息工程学院, 江西南昌 330013)

摘要: 详细讨论了利用面向对象技术的最新发展成果之一 UML(Unified Modeling Language) 及其支持工具 Rational Rose 对应用于电气铁道 SCADA 系统中实时数据库子系统的分析、设计和实现的全过程¹⁹。

关键词: SCADA 系统; 实时数据库; UML; 可视化建模工具 Rational Rose

中图分类号: TM 734; TM 922.3; TP31 **文献标识码:** A

0 引 言

SCADA (Supervisory Control And Data Acquisition) 系统是一项高技术、多学科、高难度的复杂系统, 它的开发、应用和发展与计算机技术、通讯技术和控制技术紧密相关¹⁹。由华东交通大学信息与控制工程研究所研制的微机 SCADA 系统 HY 200 正应用在衡广线郴韶段电气化铁道上, 并于 1999 年 1 月 31 日通过了铁道部技术鉴定, 得到了同行专家的一致认可和好评, 在 SCADA 领域中达到了国内先进水平¹⁹。要使 SCADA 系统的功能越来越强, 性能越来越高, 而成本越来越低, 必须把日新月异快速发展的相应技术应用到 SCADA 系统中来¹⁹。正是基于这种认识, 在原成功的基础上, 又进行了更新设计, 实时数据库子系统设计 and 实现就是其中主要且关键的一部分¹⁹。

实时数据库子系统是 SCADA 系统的核心之一¹⁹。实时数据库子系统分析设计包含实时数据库结构分析设计和实时数据库管理程序分析设计两部分组成¹⁹。在设计实时数据库子系统的过程中, 利用面向对象的方法和技术, 设计好它的类, 是实现实时数据子系统成功的关键所在¹⁹。

目前, 面向对象技术的最新发展成果之一就是统一建模语言 UML(Unified Modeling Language) 及其支持工具的出现¹⁹。UML 不仅统一了 Booch、Rumbaugh 和 Jacobson 的表示方法, 而且对其作了进一步的发展¹⁹。1997 年 11 月 17 日, OMG 采纳 UML 1.1 作为基于面向对象技术的标准建模语言¹⁹。而且继 UML 问世以后, 目前已有几种支持 UML 的可视化图形建模产品问世, 并在不断改进, 其中 Rational 公司 Rational Rose 可以说在目前的面向对象分析、建模、设计与工具市场上起着主导作用¹⁹。本篇文章将较为详细地介绍利用 UML 及其支持工具 Rational Rose 对应用于电气化铁道 SCADA 系统中实时数据库子系统的分析、设计和实现的全过程¹⁹。

1 需求与分析

实时数据库结构需满足电气化铁道 SCADA 系统的数据特点的要求;实时数据库管理程序的功能要求如图 1 所示¹⁹。

图 1 实时数据库管理程序用例图中,人状的图形表示执行着,椭圆表示用例¹⁹。由于对实时数据库具有存放内存中、对其中的数据对象是便于访问、每一个数据对象都是对应于现场的某一种实时量且要及时保持一致、系统发生异常时能够及时响应报警并作报警记录、根据实时关系表达式计算出实时关系数据库的值、当系统删除一站所时或者某一站所数据需要删除时要在实时数据库中完成对相应部分的删除等需求,因此要求实时数据库管理系统要完成图中所示的初始化、对象查找、修改更新、报警、计算、增加与删除等功能¹⁹。这些功能反映在后面设计的类的方法中¹⁹。

分析的过程也就是提取实时数据库子系统的概念层的类的过程,根据实际需要,可以初步确定五个与数据有关的类,它们是:状态量类 TStausDb, 模拟量类 TAnalogDb, 脉冲量类 TPulseDb, 状态关系量类 TStatsRelDb, 模拟关系量类 TAnalogRelDb¹⁹。属于这些类的对象分别对应于现场的每个实时量¹⁹。由于实时量很多¹⁹因而建立的对象也很多,为了能够很方便地管理这些对象,我们建立专门的类来管理它们¹⁹。TDStation 用来管理某个控站的所有实时数据库, TStation 用来管理整个系统的所有实时数据库¹⁹。这样做也很符合面向对象的思想,其出发点和基本目标是使我们分析、设计和实现一个系统的方法尽可能接近我们认识一个系统的方法,在 SCADA 系统中,我们很直观地就可以想到系统中控制和监视的是不同的站所,而每个站所中有状态量、模拟量、脉冲量等¹⁹。在分析时还要初步考虑各个类之间的关系,或提供多种连接关系供设计时选择¹⁹。

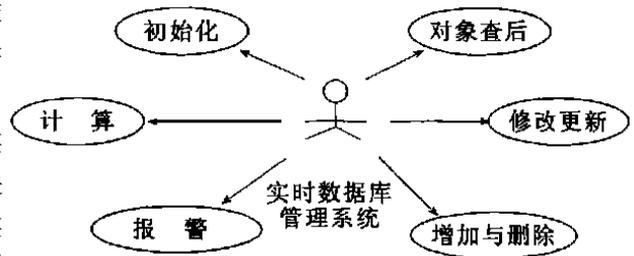


图1 实时数据库管理程序用例图

2 设计

设计时,主要是根据系统特点和要求确定类的属性和方法,并确定各个类之间的连接方式,这是一个反复并仔细推敲的过程¹⁹。我们最终定类的连接方式如图 2 所示,主要是这种方式能更好地满足系统实时性的要求¹⁹。

图 2 中首先用 TDStation 将这些实时数据各个对象聚集起来便于管理¹⁹。由于一个系统可能有很多个被控站,为了能够将所有的被控站的实时数据库统一管理起来,我们又用 TStation 来聚集管理所有 TDStation 对象,通过 TDStation 对象这个中介我们就可以管理整个实时数据库¹⁹。有了这些类,我们可以通过这些类的成员函数来统一对这些对象进行操作,使得具体的实时数据库对象对其它程序是透明的¹⁹。

从图 2 中还可以看到,其中的类比分析时多了一个 TDb 类,这是因为在具体确定各类的属性和方法时发现状态量类、模拟量类、状态关系量类、模拟关系量类具有许多共性,所以把它们共性抽象出来,形成一个基类 TDb,基类 TDb 包含了各类所共同具有的属性和方法¹⁹。这样做也给其它程序提供了访问了实时数据库的共同接口,不管它们访问的是状态量类还是模

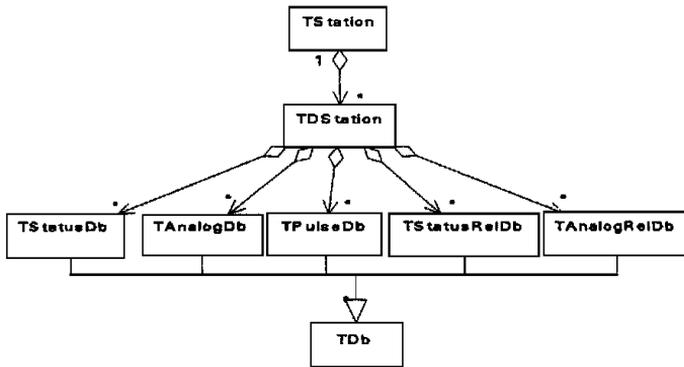


图 2 类的连接方式

拟量类,它们都可以通过访问基类 TDb 来得到¹⁹。

各个类的具体的属性和方法很多,图 2 中只显示类名,它们的属性和方法都被压缩起来了,建模工具 Rational Rose 具有这个功能¹⁹每个类应包括三部分:类名、属性和方法,以类 TDb 为例,它的类名、属性和方法如图 3 所示¹⁹。

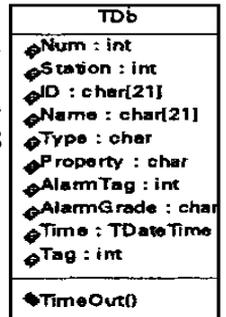


图 3 类 TDb

3 实现

在这一阶段,主要设计构件或构件图,并指明每个构件中实现的是哪些类,用什么语言实现,或在类的规格说明中指明该类属于哪个构件¹⁹然后利用建模工具的正向工程产生一个头文件和一个 cpp 文件,并在 cpp 文件中编写实现函数的具体代码¹⁹。

我们在构件视图中为实时数据库命名一个构件 db,并在 db 构件的规格说明(Specification)中指定 TStation、TDSStation、TDb、TStatusDb、TAnalogDb、TPulseDb、TstatusRelDb、TAnalogRelDb 这些类,这些类的名称左边将出现一个红色的小勾,并指定实现的语言为 C++,如图 4 所示¹⁹。

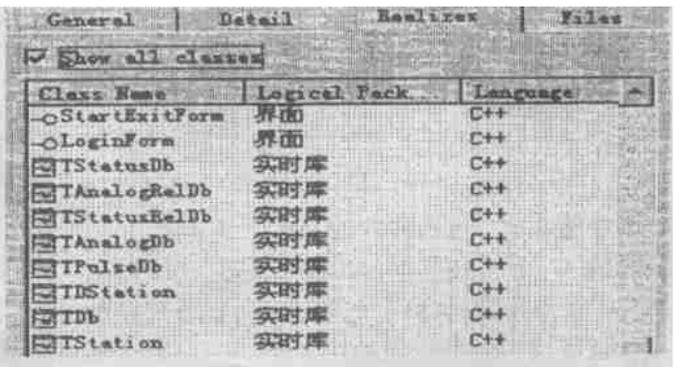


图 4 db 构件的规格说明图

然后,用右键点中 db 构件,在它的弹出式菜单中选中 Generate Code,这样代码就产生了,这样产生的代码

是头文件和 cpp 文件的代码放在一起的¹⁹如果想它们分开显示,则要把 db 构件做成两个属于不同模板的 db 构件,一个模板为 Package Specification,另一个模板为 Package Body,分别产生头文件和 cpp 文件的代码¹⁹然后利用 Tools 的下拉菜单中 C++ 的子菜单 Browse Header 和 Browse Body,就可以从记事本上分别浏览到头文件和 cpp 文件的代码了¹⁹。

由于构件中的类较多,所以产生的代码也很多,但结构都相似¹⁹。以类 TStatusDb 为例,去掉一些注释之后的头文件中的代码如下

```
class TStatusDb:public TDb
{
public://## Constructors (generated)
    TStatusDb();
TStatusDb(const TStatusDb &right);
//## Destructor (generated)
TStatusDb();
    //## Assignment Operation (generated);
    const TStatusDb &operator=(const TStatusDb &right);
    //## Equality Operations (generated)
    int operator==(const TStatusDb &right) const;
    int operator!=(const TStatusDb &right) const;
    //## Get and Set Operations for Class Attributes (genetated)
    const char get-Class () const;
    void set-Class (char value);
    const char get-CtrlEnabled () const;
    void set-CtrlEnabled (char value);
    .....
protected:
private:
private://## implementation
//Data Members for Class Attributes
char Class;
char CtrlEnabled;
int InModule;
int ImAddr;
int OutModule;
int OutAddr;
    .....
};
//## Get and Set Operations for Class Attributes (in line)
inline const char TStatusDb:get-Class () const
{
    return Class;
}
inline void TStatusDb:set-Class (char value)
{
    Class=value;
}
    .....
```

在 C++ Builder 中产生一个 New Unit,把它存为 db,把自动产生的代码稍加修改后进行拷贝和张贴,然后把要实现的具体函数的代码写入¹⁹。

4 结束语

中国知网 <https://www.cnki.net>

由于有 HY²⁰⁰ 微机 SCADA 系统成功的经验,再加上在本次设计过程中应用最新面向对

象技术,使得实时数据库子系统的设计既缩短了时间又提高了成功率,目前已在实验室与 SCADA 系统的其他程序一起调试成功¹⁹希望本文能起到抛砖引玉的作用,并能对面向对象的方法和技术设计程序的读者有一定的启发¹⁹.

[参 考 文 献]

- [1] UML Notation Guide·<http://www.rational.com/uml>¹⁹.
- [2] Rational Rose help on line·<http://www.rational.com/prod/rose/index.jhtml>
- [3] 钱清泉主编¹⁹微机临近系统原理[M]¹⁹.北京:中国铁道出版社,1996,6¹⁹.
- [4] 陈剑云,杨丰萍¹⁹.HY200 电气铁道 SCADA 系统[J]. 电力自动化设备,2000,(1):28~30¹⁹.

The Object-Oriented Modeling of the Sub-system Real-time Database in the SCADA System

YANG Feng-ping

(School of Electrical and Information Engineering, East China Jiaotong University, 330013, China)

Abstract: This paper discusses the whole process of analyzing, designing and implementing the subsystem real-time database in the SCADA system used on electrified railways by applying the UML(Unified Modeling Language) , one of the latest development achievements of object-oriented technology, and its visual modeling tool Rational Rose.

Key words: SCADA systems; real-time database; UML; visual modeling tool Rational Rose