

# 基于.NET的WebService自动站数据服务平台的设计与实现

## Design and Implementation of Automatic Station Data Services Platform Basing on WebService of .NET

黄 志, 高钰杰, 任思宇

HUANG Zhi, GAO Yu-jie, REN Si-yu

(广西区气象信息中心, 广西南宁 530022)

(Guangxi Meteorological Information Center, Nanning, Guangxi, 530022, China)

**摘要:** 为了进一步做好气象资料的开发应用和共享服务工作, 实现对自动站各气象要素日、月和年统计数据的快速获取, 基于.NET的WebService自动站数据服务平台通过在数据库服务器端构建存储过程, 配合基于.NET平台构建WebService数据接口, 将数据对象的内容转换成标准的XML格式, 同时支持以网页下载的形式获取数据。基于.NET的WebService自动站数据服务平台运行高效稳定并在实践中取得了很好的效果。

**关键词:** 服务平台 气象数据 WebService .NET 存储过程 XML

中图法分类号: TP393.09 文献标识码: A 文章编号: 1002-7378(2012)04-0261-03

**Abstract:** For the further development and application of meteorological information and its shared services, an automatic station data services platform based on WebService of .NET was established to obtain statistical data of days, months and years about all meteorological elements from the automatic weather station. Store procedure was built through database server porter. With the help of WebService data interfaces based on .NET platform, the contents of data object will be converted into the standard XML format and support web page download. Platform is proved to run stably and effectively with good results in practice.

**Key words:** service platform, meteorological data, WebService, .NET, stored procedure, XML

随着社会和经济的不断发展, 气象数据也发挥越来越重要的作用, 特别是自动气象站的小时实时多要素数据作为气象数据的基础数据更是在气象数据共享服务中占据着重要的地位, 许多部门对于温度、降水、气压及湿度等几个要素的日、月和年统计数据的需求日益常态化。为了进一步做好气象资料的开发应用和共享服务工作, 加强资料管理、服务单位与应用单位或人员的交流, 充分发挥气象资料在提高气象业务、科研、服务能力的基础性作用, 我们搭建了一个高效、安全的基于.NET的WebService

自动站数据服务平台。

### 1 平台开发技术概述

Webservie 是一种面向服务结构(SOA)的具体实现, 它的平台无关性可以支持各类编程语言和硬件平台, 并集成软件应用程序。WebService 是一个应用组件, 各应用程序和通过网络协议和规定标准数据格式(HTTP, XML, SOAP)来访问 WebService, 通过 WebService 内部执行得到所需结果。WebService 涉及的相关技术主要包括可扩展标识语言(XML)、简单对象访问协议(SOAP)、Web 服务描述语言(WSDL)、通用描述、返现与集成(UD-DI)。其中, XML 是 WebService 平台中表示数据的基本格式; SOAP 通过 XML 文档形式发送和接收消息, 实现在异构平台上的不同系统的相互通信

收稿日期: 2012-08-06

修回日期: 2012-08-25

作者简介: 黄 志(1981-), 男, 硕士, 工程师, 主要从事软件设计与信息服务工作。

和数据共享;WDSL 用来描述 WebService 所完成的功能和 WebService 所提供的服务;UDDI 用来注册、发布已建立的 WebService<sup>[1,2]</sup>。

.NET 是微软的新一代技术平台,为敏捷商务构建互联互通的应用系统,这些系统是基于标准的、联通的、适应变化的、稳定的和高性能的。从技术的角度,一个 .NET 应用是一个运行于 .NET Framework 之上的应用程序。更精确的说,一个 .NET 应用是一个使用 .NET Framework 类库来编写,并运行于公共语言运行时 Common Language Runtime 之上的应用程序。.NET 用来实现 XML, Web Services, SOA(面向服务的体系结构 service-oriented architecture) 和敏捷性的技术。Visual Studio .NET 是一套完整的开发工具,用于生成 ASP Web 应用程序、XML Web services、桌面应用程序和移动应用程序。Visual Basic .NET、Visual C++ .NET、Visual C# .NET 和 Visual J# .NET 全都使用相同的集成开发环境 (IDE),该环境允许它们共享工具并有助于创建混合语言解决方案。另外,这些语言利用了 .NET Framework 的功能,此框架提供对简化 ASP Web 应用程序和 XML Web services 开发的关键技术的访问<sup>[3]</sup>。

### 2 平台的设计与实现

由于自动站数据为整点小时数据,需要通过编写相应软件按地面气象观测规范处理生成合格的气象要素日数据,为之后统计合并月、年数据的快速提取提供基础保证。因此在搭建此平台之前,需要一套后台程序定时提取日数据到中间表中。表 1 为中间表部分表结构,图 1 为平台的系统框架<sup>[4]</sup>。

表 1 自动站要素日数据部分表结构

气象要素列名	数据类型	备注
Stanum	Varchar(5)	站号
Stadate	Int	日期
Maketime	Int	起报时间
Avgtemp	Float	日平均气温
Mintemp	Float	日最低温度
Maxtemp	Float	日最高温度
Rain	Float	日累计雨量
...	...	...

#### 2.1 服务器端设计

服务器端设计主要是自动站日数据处理、建立存储过程和构建基于 .NET 平台的 WebService 应用数据接口。自动站数据主要是以小时整点数据为主,只有按照相关文件要求加工整合成日数据,日数

据合并为月以及年数据,才能方便用户的快速查询和提取数据,为此需要做个后台程序定时对实时小时自动站数据做加工处理,并生成中间表存储日、月和年数据。建立存储过程可以提高查询速度,对于一个面对全省的通过广域网传输的数据接口,它可以减少网络通信量,将工作负荷加载到服务器端,简化客户端工作量并具有较好的安全性,同时方便开发人员随时升级和拓展代码功能,前台应该程序和数据库代码可以分别独立进行,加快开发改进的速度。构建基于 .NET 平台的 WebService 应用数据接口只需在框架内编写相应的功能代码即可,因为 .NET 框架有专门的代码封装好的 WebService 的应用框架。

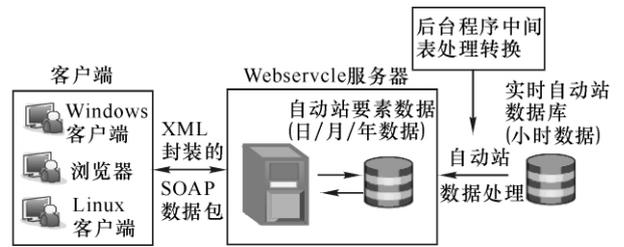


图 1 平台系统框架

为了方便客户获取数据,此平台设计的 WebService 接口生成的是标准 XML 结构数据,比 WebService 默认接口生成的 XML 数据更为简洁明了。以下是部分关键代码。

```

[WebMethod (Description=" 直接转换为标准 xml 格式" )]
public XmlDataDocument GetXmlAData ( string
stanum, string btime, string etime, string make-
time,int rain,int ymd)
{
    DataSet dsNorthwind = new DataSet();
    InitializeComponent ( stanum, btime,
etime,maketime,rain,ymd);
    .....
    XmlDataDocument doc = new XmlDat-
aDocument(dsNorthwind);
    doc.Save(Console. Out);
    .....
    XmlNode root1 = doc. Document-
tElement;
    XmlNodeList roots = root1. Select-
Nodes(" list");
    foreach (XmlNode roota in roots)
//为所有元素加上站点名称标记

```

```

    {
        XmlElement Link = doc. CreateElement(" SiteName" ); // 转成 XML 数据
        Link. InnerText = ConfigurationSettings. AppSettings[" SiteName" ]. ToString();
        .....
    }

```

### 2.2 客户端程序设计

目前客户端程序可以通过直接调用 WebService 接口或是构建 .NET 网页调用 WebService 接口 2 种方式获取数据,通过对站点、时间和气象要素等参数配置可以快速获取其数据产品。

#### 2.2.1 直接调用 WebService 接口

WebService 接口代码编程完成,编译之后生成可以通过网站发布的形式直接调用 WebService 接口,输入参数即可获取数据。表 2 为 WebService 提供的数据接口参数。图 2 为相应的调用和结果回执界面。

表 2 WebService 提供的数据接口参数

Stanum	站点参数(可多站点,用逗号分隔)
Btime	统计开始时间(yyyymmdd/年月日)
Etime	统计结束时间(yyyymmdd/年月日)
Maketime	起报时间(08 时或 20 时)
element	统计要素(1 代表雨量,2 代表温度,···,可多选)
ymd	统计年、或月、或日数据,1 代表年,2 代表月,3 代表日



图 2 相应的调用和结果回执界面

#### 2.2.2 通过 .NET 网页调用 WebService 接口

基于 .NET 平台建立网页,应用可视化界面,通过存储过程调用 WebService 接口并填充数据到相应的控件显示,极大地增强了交互性。图 3 为通过 .NET 构建的数据下载界面,生成压缩文件供用户直接下载使用。相关的代码如下。

```

localhost. WebService np = new localhost. WebService(); // 生成 Webservice 接口对象,并获

```

### 取数据集

```

DataSet ds = np. GetData(StanumSql, Sbe-
gin, Send, maketime, SelectedItem. Value, ToString
(), rain, ymd);
.....
// 将数据集加工并生成 .rar 压缩文件
Reg = Registry. ClassesRoot. OpenSubKey
(@" Applications \ WinRAR. exe \ Shell \ Open \
Command" );
.....
StartInfo = new ProcessStartInfo();
StartInfo. FileName = Rar;
StartInfo. WindowStyle = ProcessWindowStyle.
Hidden;
StartInfo. WorkingDirectory = SourcePath;
Process = new Process();
Process. StartInfo = StartInfo;
Process. Start();

```



图 3 通过 .NET 构建的数据下载界面

2 种调用 WebService 接口方式的运行效果都非常良好。直接调用 WebService 接口获取数据显得非常简洁,显示的数据为标准的 XML 数据结构,但是对于数据的提取不太方便;通过 .NET 构建网页并下载获取数据则方便快捷,可视化效果较好,但是增加了程序设计的复杂度。

### 3 结束语

使用 Webservice 实现信息化网络数据共享传输具有较为明显的优势,可以将大量复杂操作集中在服务器端完成,而客户端程序可以象调用本地函数一样调用 WebService 接口函数,简化了客户端程序编写的难度,可跨多个平台,充分体现了异构平台下的分布式开发优势。基于 .NET 的 WebService 自动站数据服务平台在实际的测试中可以高效快速的获取所需的气象要素和气象统计数据(标准

(下转第 276 页)

```

DelayMS(300);
YELLOW __A = ~ YELLOW __A;
GREEN __A=1;
if(++Flash_Count !=10) return; //
闪烁 5 次
Flash_Count=0;
Operation_Type=3;
break;
case 3://东西向红灯,南北向绿灯亮
RED __A=0;YELLOW __A=1;GREEN __
__A=1;
RED __B=1;YELLOW __B=1;GREEN __
B=0;
DelayMS(2000);
Operation_Type=4;
break;
case 4://南北向黄灯闪烁 5 次
DelayMS(300);
YELLOW __B = ~ YELLOW __B;
GREEN __B=1;
if(++Flash_Count !=10) return;
Flash_Count=0;
Operation_Type=1;
}
}
//主程序
void main()

```

```

{
while(1) Traffic_Light();
}

```

## 6 结束语

通过利用 Proteus 软件仿真上述的交通灯控制系统,实验测试结果证明用本系统单片机能完成交通灯控制过程,有效地疏导交通,提高了交通路口的通行能力。采用 Proteus 仿真软件进行虚拟实验,具有非常明显的优势,无需涉及和消耗电子元件即可设计好电路图,并可详尽了解该电路的元件的参数及电路的性能。实验内容全面、硬件投入少、省时省力、实验数值比较可信等。电路仿真成功后就可以做 PCB 板及后续工作。

### 参考文献:

- [1] 曲立国,黄友税,唐超礼,等. Proteus 在电子课程设计中的应用[J]. 科技视界,2012(14):96-98.
- [2] 从宏寿. 电子设计自动化——Proteus 在电子电路与 51 单片机中的应用[M]. 西安:西安电子科技大学出版社,2012.
- [3] 朱定华. 电子电路实验与课程设计[M]. 北京:清华大学出版社,2009.
- [4] 冯乔,李冬. Proteus 在模拟电子实践教学中的应用[J]. 信息技术,2011(31):38-40.

(责任编辑:邓大玉)

(上接第 263 页)

XML 格式),在之后的工作中可加入湿度、气压以及风速等几个气象要素数据,完善数据结构;并且可以构建基于不同数据产品和不用用户的 Webservice 数据接口,在注重数据安全的同时不断提高数据服务的能力。

### 参考文献:

- [1] 吕宏伟. SOA 体系结构中的 Webservice 技术[EB/LO]. [2012-07-26]. <http://wenku.baidu.com/view/6cf6260b4a7302768e993902.html>.

- [2] 动态网站制作指南. 可扩展标记语言 XML、Web 服务和 .NET 框架(2)[EB/LO]. [2012-07-26]. [http://www.ccidedu.com/art/1925/20041027/170047\\_1.html](http://www.ccidedu.com/art/1925/20041027/170047_1.html).
- [3] 百度百科. NET. [2012-07-26]. <http://baike.baidu.com/view/4294.htm>.
- [4] 中国气象局. 地面气象观测规范[M]. 北京:气象出版社,2003:121-125.

(责任编辑:邓大玉)