

Implementation of the AutoComplete Feature of the Textbox Based on Ajax and Web Service

Zhiqiang Yao

Dept. of Computer Science, North China Institute of Aerospace Engineering, Langfang, China
E-mail:yaonciae@126.com

Dr. Abhijit Sen Ph.D.

Dept. of Computer Science & Information Technology, Kwantlen Polytechnic University, Canada
E-mail: abhijit.sen@kwantlen.ca

Abstract—Textbox input is widely used on the web for user input. In order to provide convenience for the web users and increase the input efficiency, one can use AutoComplete feature with the textbox. AutoComplete feature predicts possible word matches for entries that begin with the prefix typed into the textbox without the user actually typing the whole word completely. This paper describes the design and implementation of an ASP.NET application based on AutoComplete Extender control from AJAX Control Toolkit. The textbox control is bound with the AutoComplete Extender and then web service is deployed to connect the database server that stores the possible words. User keystrokes are monitored and textbox input is completed based on first typed letters that match with data stored in database. An AutoComplete test program is developed and tested.

Index Terms—AutoComplete, Web service, Ajax, Database, ASP.NET

I. INTRODUCTION

AutoComplete [1] is a feature provided by many web browsers, e-mail programs, search engine interfaces, source code editors, database query tools, word processors, and command line interpreters. AutoComplete enables the program to predict a word or phrase that the user wants to type in without the user actually typing the word completely. The word being typed can be easily predicted if list of words is stored in a database. AutoComplete speeds up human-computer interactions.

This article presents the design and implementation of AutoComplete application using AutoCompleteExtender from ASP.NET [2] AJAX [3] Control toolkit and is used to display and select possible matches for entries as user types in letters in a textbox control. We have also presented the design of the database using Microsoft SQL

Server [4], to store and retrieve data as requested by other software applications [5]. Microsoft Visual Studio 2010[6] integrated development environment (IDE) is used to develop applications using C# programming language for the auto completion of the textbox.

II. COMPONENTS OF AUTOCOMPLETE APPLICATION

AutoComplete feature can be realized using widely used Ajax (Asynchronous JavaScript and XML [7]) web application development framework. The Ajax based web application model [8] is shown in the Figure 1.

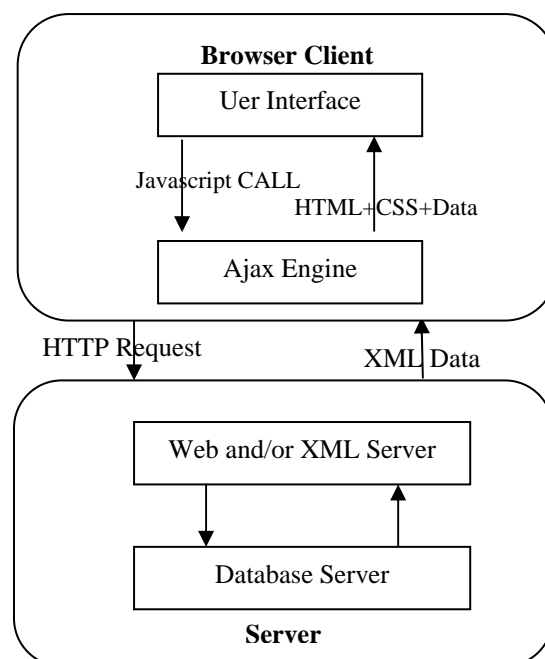


Figure 1 Ajax-based Web Application Model

A. AutoCompleteExtender (Ajax Control)

AutoCompleteExtender from the Ajax control tool box can be bound with the asp.net textbox control in the web pages. The Ajax AutoCompleteExtender control can be configured to call the web method created inside a web

Manuscript received October 1, 2012; revised January 8, 2013; accepted January 28, 2013.

Project Number: No.2012011021 of Langfang Scientific Technology Bureau, Hebei province

Corresponding author: Zhiqiang Yao

Dept. of Computer Science, North China Institute of Aerospace Engineering, Langfang, China 065000 E-mail:yaonciae@126.com

service. In response to data query, a web method can respond with the requisite data for the AutoComplete. AutoCompleteExtender attached to any TextBox control, will associate that control with a popup panel to display possible candidate words. These candidate words are provided by web service method that begin with the prefix typed into the textbox. The user can then select appropriate candidate words from the dropdown menu.

B. Creation of e Web Service

Web services [9] are components on a Web server that a client application can call by making HTTP requests across the Web.

ASP.NET Web services are called from client script that runs in AJAX-enabled ASP.NET Web pages.

One web method is created as a web service. A client script can be configured to call this web method.

Before the application of the web service, one can test the web service separately by running the web service on the internet explorer.

C. Database Server Deployment

Microsoft SQL Server 2000[10] is used to design the database: schoolinfo .The example database contains a single table tb_dept which stores the related data for AutoComplete. This table stores the information of the departments' name of one school.

The structure of the table is shown in table I, where the deptName column is an indexable column.

TABLE I.
DESIGN OF THE DATA TABLE :TB_DEPT

| ColumnName | Data Type | Remarks |
|------------|-------------|---|
| deptID | int | Automatic Increase by 1.ID for table data |
| deptName | Varchar(30) | Department name,indexable |

III. DEVELOPMENT OF THE PROGRAM

Dynamic applications [11] are applications that build content “on the fly” in response to request made. They can reflect the interaction result on the web between the client and the server. Visual Studio 2010 development tool is used to design the web page and publish one web service to fetch data from the database server. In order to implement Ajax AutoCompleteExtender control dynamic link library file the AjaxControlToolkit.dll from the Ajax control toolkit, is included in the ASP.NET bin folder.

A. Conceptual Software Architecture

The conceptual software architecture [12] is shown in Figure 2 and the architecture contains the following components:

- Database server [13], which serves the data storage, accepts the requests from the web service and sends the responses to the web service.

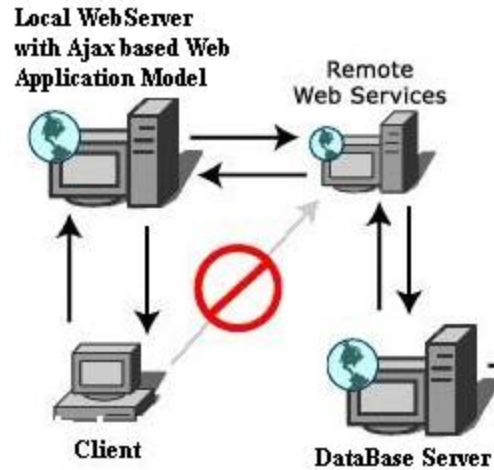


Figure 2 the conceptual architecture of AutoComplete system

- Web server, which is realized by the configuration of the Internet Information Server (IIS), gets data directly from the web service. ASP.NET framework is used to develop web application which sends requests to the web service and accepts the responses from the web service and delivers data to the client (PC, Phone, etc.). The AutoComplete test program is deployed on the web server.

- Web service, which sends requests to the database server and gets the response of the data query result from the database, is responsible to get connected with the database server. It depends upon the Ajax extender, which is applied in the ASP.NET framework of the web server.

B. Web Page Design

The system is designed using With Visual Studio 2010. The source code of the autocomplete.aspx web page is shown in Appendix A and Figure 3 shows the layout of the autocomplete.aspx web page. It contains the following controls:

- an ASP.NET TextBox control for inputting department name
- an Ajax extensions control ScriptManager for managing java script codes and allowing the use of the Ajax Toolkit of AutoCompleteExtender
- an Ajax Toolkit of AutoCompleteExtender attached to an ASP.NET TextBox control to get AutoComplete behavior.



Figure 3 Layout of the autocomplete.aspx web page

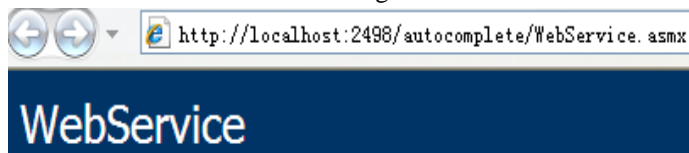
C. Publishing one Web Service

Web service WebService.asmx is added to the AutoComplete website and the source code for this web service is shown in Appendix A.

The web service [14] consists of getCompleteDept web method, which is invoked to get the department data from the database server.

One SQL clause: select dept_name from tb_dept where dept_name like "" + prefixText + "%' order by dept_name is used to filter words from the database server. The like clause is composed of one constant string: prefixText and one wildcard character %.

Figure 4 shows the running of the web service and the execution of the web method getCompleteDept. As shown in Figure 5, the web method getCompleteDept needs two input parameters prefix Text (which is set to 'Te') and count (set to 10). The CALL button is pressed to test the web method and Figure 6 shows the test result



- [getCompleteDept](#)

Figure 4 Test of the web service

with one xml [15] file containing the data fetched from the database server.

D. Test of the Program

The configuration of the IIS (Internet Information Services) [16] is shown in Figure 7, which contains autocomplete virtual directory. The autocomplete virtual directory includes the following web files:

- autocomplete.aspx,
- autocomplete.aspx.cs,
- WebService.asmx,
- App_Code/WebService.cs,
- bin/ AjaxControlToolkit.dll and
- web.config

With the configuration of the web server and database server, the AutoComplete system is tested by populating table tb_dept with the sample department data shown in the Appendix B. The test result is shown in Figure 8 and Figure 9. After inputting 2 letters, the related data is shown automatically in a list from where one can select the appropriate item. As the user types the first alphabet, in the sample information system, i.e., the department name should be typed in, the words from the database are extracted and further narrowing down of words occur as user types more alphabets (shown in Figure 10).

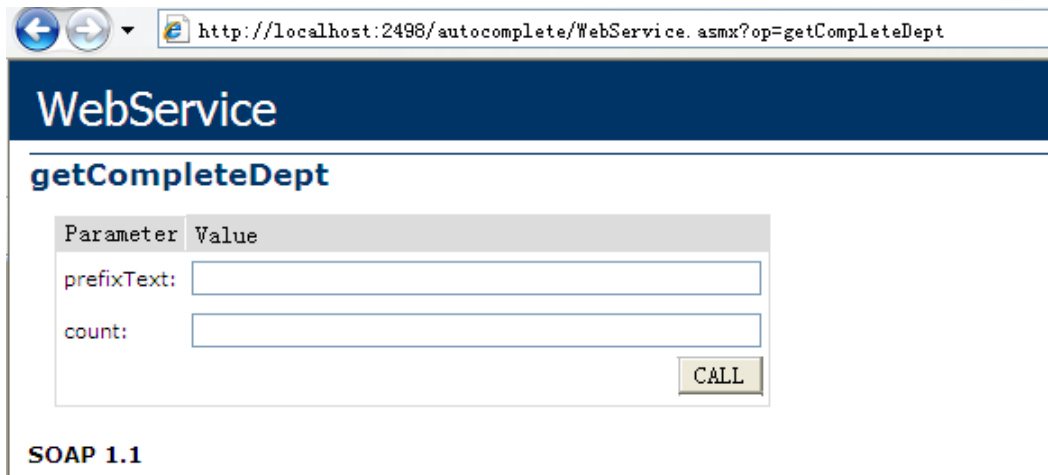


Figure 5 Test of the web method in the web service

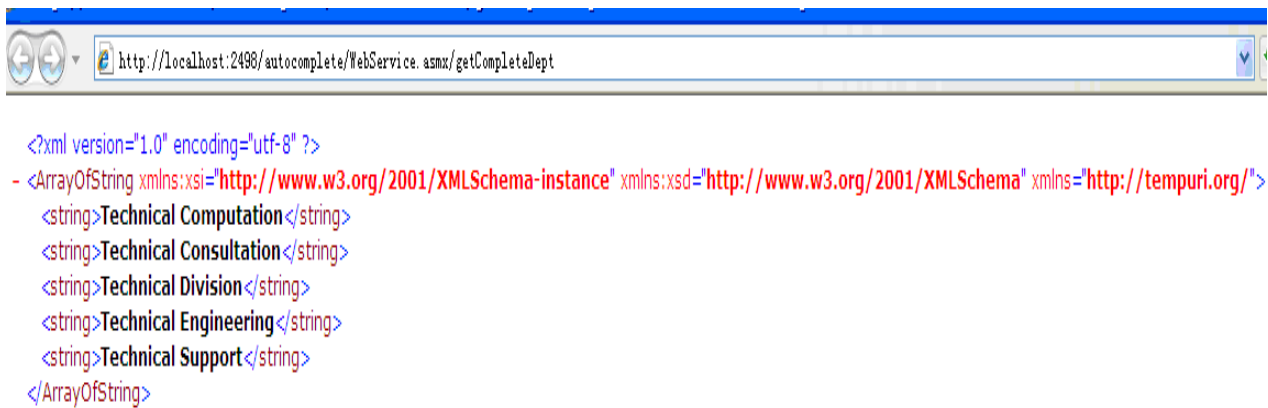


Figure 6 Testing result of the web method in the web service

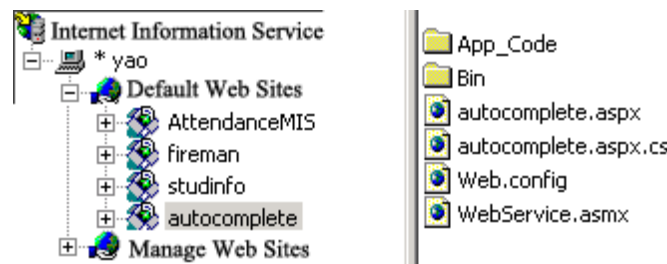


Figure 7 the configuration of IIS

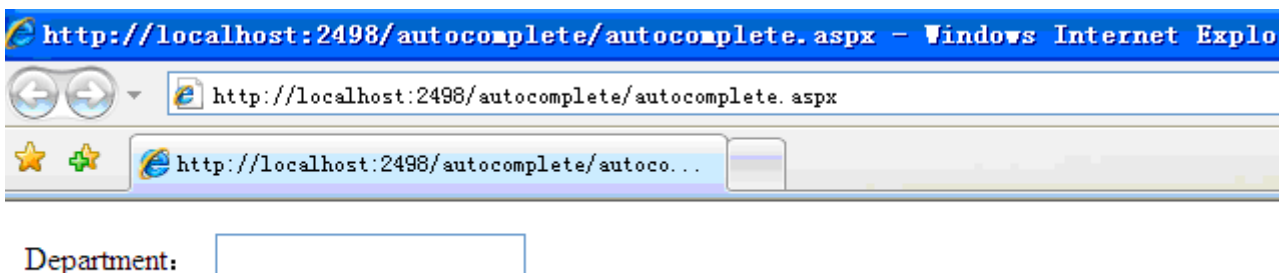


Figure 8 Test of the program (Before inputting 2 letters)

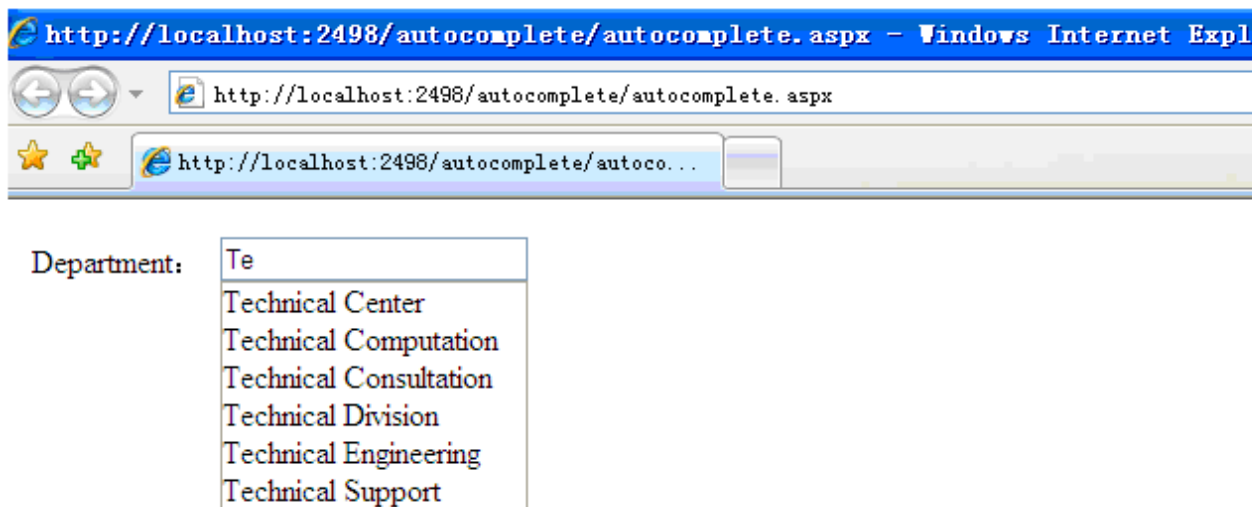


Figure 9 Test of the program (After inputting 2 letters)

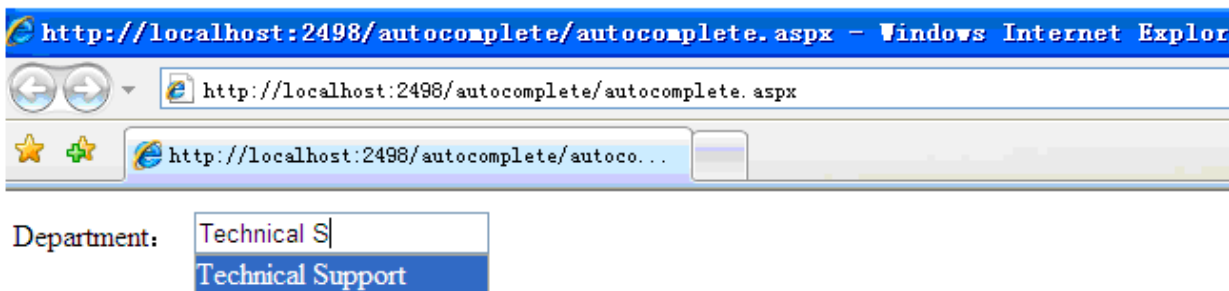


Figure 10 further narrowing down of words occur as user types more alphabets

IV. CONCLUSION

In this paper, we have discussed the design and implementation of the textbox autocomplete test programs. We have proposed a solution based on the web service [17], using Microsoft SQL server database and AutoCompleteExtender feature of Ajax toolkit control. A benefit to our approach is that with the realization of the autocomplete, the user input efficiency is highly improved and unnecessary spelling error can be avoided and human computer interactions [18] are strengthened.

The correct prediction of words depends upon the extensiveness of the database. Before providing the autocomplete function of the textbox input, we should initialize the table data of the database and make it cover almost the whole possible words.

This conceptual design framework can be easily integrated in any other web applications [19].

APPENDIX A SOURCE CODES OF THE PROGRAM FILES

File: *autocomplete.aspx*

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="autocomplete.aspx.cs"
Inherits="autocomplete" %>
<%@ Register Assembly="AjaxControlToolkit"
Namespace="AjaxControlToolkit"
TagPrefix="ajaxToolkit" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-
transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server"> <title></title></head>
<body> <form id="form1" runat="server"> <div>
<asp:ScriptManager ID="ScriptManager1"
runat="server"> </asp:ScriptManager>
Department:<asp:TextBox ID="txt_dept"
runat="server"></asp:TextBox><ajaxToolkit:AutoComple
eteExtender ID="ac1" runat="server" Enabled="true"
TargetControlID="txt_dept"
ServicePath="WebService.aspx"
ServiceMethod="getCompleteDept"
MinimumPrefixLength="2"
CompletionSetCount="10"></ajaxToolkit:AutoComplete
Extender> </div> </form></body></html>
```

2. File: *WebService.aspx*

```
WebService.aspx
<%@ WebService Language="C#"
CodeBehind="~/App_Code/WebService.cs"
Class="WebService" %>
```

3. File: *WebService.cs*

```
App_Code/WebService.cs
```

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Configuration;
using System.Data.SqlClient;
using System.Data;
[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo =
WsiProfiles.BasicProfile1_1)]
// ASP.NET AJAX calls this web service from the
script code
[System.Web.Script.Services.ScriptService]
public class WebService :
System.Web.Services.WebService
{
public WebService ()
{
}
private static string[] autoCompleteWordList = null;
[WebMethod]
public string[] getCompleteDept(string prefixText,int
count)
{
if (autoCompleteWordList == null) { SqlConnection
conn = new SqlConnection();
conn.ConnectionString =
ConfigurationManager.ConnectionStrings["connstr"].ToS
tring();
conn.Open();
SqlDataAdapter da = new SqlDataAdapter("select
dept_name from tb_dept where dept_name like "" +
prefixText + "%' order by dept_name", conn);
DataSet ds = new DataSet(); da.Fill(ds);
string[] temp = new string[ds.Tables[0].Rows.Count];
int i = 0; foreach (DataRow dr in ds.Tables[0].Rows)
{ temp[i] = dr["dept_name"].ToString();i++;
}
autoCompleteWordList = temp;
if (conn.State == ConnectionState.Open)
conn.Close();
}
string[] returnvalue = new string[count];
returnvalue = autoCompleteWordList;
return returnvalue;
}
}

```

APPENDIX B SAMPLE DATA FOR THE PROGRAM TESTING

Sample Data of the Table: tb_dept is shown in the table II.

ACKNOWLEDGMENT

The author, Zhiqiang Yao wishes to thank all the reviewers for their valuable comments and suggestions that improved the quality of this paper; what's more, he wishes to thank Abhijit Sen for his good advice on updating this paper and for his necessary support in

teaching and scientific research. Thanks a lot for his great cooperation in this paper.

This research is supported by the project (No.2012011021) of Langfang Scientific Technology Bureau, Hebei province.

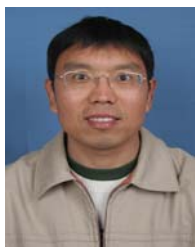
TABLE II. SAMPLE DATA OF TABLE: TB_DEPT

| deptID | deptName |
|--------|------------------------|
| 1 | Technical Division |
| 2 | Technical Engineering |
| 3 | Technical Computation |
| 4 | Technical Support |
| 5 | Technical Consultation |
| 6 | Technical Center |
| 7 | Finance Department |
| 8 | Finance Office |
| 9 | Finance Center |

REFERENCES

- [1] Wang Bo, Wang Han-bo, "Implementation of AutoComplete Based on JQuery", *Journal of SanMenXia Polytechnic*. 2010, 09(3) pp102-126
- [2] SUN Xue-hua, ZHOU Bin, WANG Hui , "Implementation of Enterprise Plan Management System Based on ASP.NET", *Journal: Automation & Instrumentation*, 2011, 36(8) pp6-8
- [3] Bruce Perry, *AJAX HACKS*, O'REILLY Media, Inc. Cambridge, 2006.5.
- [4] Tobias Thernstrom, Ann Weber, Mike Hotek, *MCTS Self-Paced Training Kit (Exam 70-433): Microsoft® SQL Server® 2008—Database Development*, Microsoft Press, February 2009, pp: 306-453.
- [5] Jennifer Niederst Robbins, *Web Design in a Nutshell (Third Editon)*, O'REILLY Media, Inc. Cambridge, 2006.5.
- [6] Scott Guthrie, Scott Guthrie: *Visual Studio 2010 and .NET Framework 4.0* , *DotNET developer community*, AFAS Theatre, Leusden, The Netherlands , September 25 2009.
- [7] Michael J. Young, *Microsoft Step By Step XML*, Oct. 2008 pp.225-276.
- [8] Jesse Liberty, Dan Hurwitz, *Programming ASP.NET (Third Editon)* ,O'REILLY Media, Inc. Cambridge, 2006.4.
- [9] WU Hai-bo, WU Bao-guo, "The Method of Accessing Multi-Source Database Based on Websevice", *J. Computer Knowledge and Technology*. 2009, 5(31).
- [10] Gou Lingyi, Chen Xiaobo, Xiong Guangleng, "e Integration of XML and Relational Data Based on SQL Server 2000", *Computer Engineering and Application*, 2001, 37(10)
- [11] Yunzhi Peng, "Efficient Sports Websites Evaluation System Based on ASP Technology", *Journal of Software*, Vol 7, No 7 (2012), 1617-1624, Jul 2012
- [12] YAO Zhiqiang, Abhijit Sen, XIA Han, "Dynamic Financial Application Framework For The Mobile Phones", *2010 2nd IEEE International Conference on Information and Financial Engineering ICIFE 2010* , Chongqing, China. Sep. 17-19, 2010.

- [13] Yu qian, Luo jing , “Using ADO.NET To Administer Database In ASP.NET”, *Microcomputer Applications*, 2004, 25(3), pp 372-376
- [14] Jianbo Bai, Yuzhe Hao, Guochang Miao, “Integrating Building Automation Systems based on Web Services”, *Journal of Software*, Vol 6, No 11 (2011), 2209-2216, Nov 2011
- [15] Haiping Xu, Abhinay Reddyreddy, Daniel F. Fitch, “Defending Against XML-Based Attacks Using State-Based XML Firewall”,
- [16] Mitch Tulloch, *Inside IIS 6*, O'REILLY Media, Inc. Cambridge, 2004.3
- [17] Guangjun Guo, Fei Yu, Zhigang Chen, Dong Xie, “A Method for Semantic Web Service Selection Based on QoS Ontology”, *Journal of Computers*, Vol 6, No 2 (2011)
- [18] Effy Oz , *Management Information System Six Edition*, Course Technology, January 2008, pp.208-254
- [19] Haiwang Cao, Chaogai Xue, Lili Dong, “Research on Approximate Reconfiguration of Enterprise Information System based on Formal Representation”, *Journal of Software*, Vol 6, No 11 (2011) ,pp 2173-2177



Zhiqiang Yao received the 4-year bachelor degree of Mechanical Engineering from Xi'an JiaoTong University in 1994 and earned the 3-year master of Hydraulic Engineering from Tsinghua University in 2000. His major field of study is on computer programming and computer application. He was an MCSD (Microsoft Certified

Solution Developer).

He has 4 years experiences of working in IT companies and during that time, he was responsible for developing the e-learning systems. In 2003, he worked at the Research Institute of the Armed Police Force Academy and did several scientific research projects. In 2009, he began to work at Computer Science Department of North China Institute of Aerospace Engineering, and he taught the First Course of Database, ASP.Net Programming, and XML and so on.

Associate Professor Yao published more than 10 papers, 4 of which were indexed by EI.



Abhijit Sen holds a PhD. from McMaster University, Hamilton, Ontario, Canada and Master of Science degree from University of California, Berkeley, USA, B.Tech in Electrical Engineering from Indian Institute of Technology, Kharagpur, India. He has over 25 years of academic and administrative experience at Kwantlen Polytechnic University. He

served as a chair of the department for over 14 years. He is currently the Professor of the Computer Science and Information System at Kwantlen Polytechnic University, BC, Canada. Abhijit also was a visiting professor at Waikato University, Hamilton, New Zealand, and Munich University of Applied Sciences, Germany, Centre for Development of Advanced Computing, India, and North China Institute of Aerospace Engineering, China.

His current research interests are in the areas of Wireless Security, Radio Frequency Identification (RFID), Computing Education and Teaching Methodologies, Distributed Systems and Databases.