A Scalable Application Architecture for Composing News Portals on the Internet

Serpil Tok and Zeki Bayram
Department of Computer Engineering
Eastern Mediterranean University
Famagusta, Turkish Republic of Northern Cyprus
serpil.tok@emu.edu.tr, zeki.bayram@emu.edu.tr

Abstract- We report on a scalable application architecture for building news portals on the Internet. The main component of this architecture is XML Web services that provide access to the articles of each author, who maintains his/her pool of articles using a desktop program. Articles are kept in different databases for each author. Various news portals can then retrieve articles from the article repositories by using an XML Web services interface. This results in tremendous flexibility in composing content for news portals and personalizing it according to user preferences. Furthermore, authors of articles benefit from the availability of powerful desktop editing tools for maintaining their repository of articles. We demonstrate the utility and feasibility of this architecture through a proof-of-concept system implemented on the .NET platform using the C# programming language.

I. Introduction

XML Web services have opened new possibilities for composing applications from software components with well specified interfaces on the Internet. Portals, on the other hand provide a combination of "out of the box" and custom functionality to allow users to find, manage, categorize, and use content and applications [1]. News portals such as Google [2] and Yahoo [3] get their content from various news sources and present them to the user in a categorized manner. The user also has the option of making advanced searches for articles.

Considering the functionality requirements of portals, it is an easy conclusion that XML Web services technology could be of tremendous help in obtaining content for portals. In this paper, we describe a news portal architecture and application based on this idea. In our architecture, news portals obtain their content material from different sources by exclusively using XML Web services. The sources are article depositories, each one residing in a database, with an XML Web services “front end”. Each repository of articles belongs to an author, who maintains it using an authoring tool in the form of a desktop application. News portals then retrieve articles from authors by using the corresponding Web service.

The biggest advantage of our architecture is that it scales very well to an increased number of sources. Furthermore, it eliminates coordination problems between publishers and authors. Each author can act independently, maintain his articles, and anyone who wishes to use his articles can do so by using his Web service. We thus envisage many news sources “popping up” on the Internet (for example freelance local reporters) and virtual “newspapers” (i.e. news portals) accessing their writings and presenting them to Internet users in a format they desire.

Another advantage is that different components can be upgraded without any interruption of the system (for example, the authoring tool, specific database used, or even the Web services themselves).

The rest of paper is organized as follows. Section II describes the overall system architecture. Section III gives a detailed description of the authoring tool used by authors of articles. Section IV reports on the “Article Server Web Service”, a Web service that provides access to articles. Section V describes the “News Portal Web application”, a small, “bare-bones” news portal that makes use of Web services to obtain and customize its content. Section VI talks about related work and finally section VII is the conclusion and future research directions.

II. The system architecture

Figure 1 illustrates the overall system architecture, where only one news portal is involved. We explain each component separately below.

The *news portal* is implemented as .NET Web application (with a .aspx extension). This portal obtains the users’ preferences, and searches the various article repositories using Web services to find and retrieve articles which meet the criteria specified by the users. These articles are then presented to the users in the form of a dynamically constructed web page.

XML Web services for each article repository provides methods for accessing, quering and manipulating of the database. Each XML Web service provides methods to access a different database of articles.

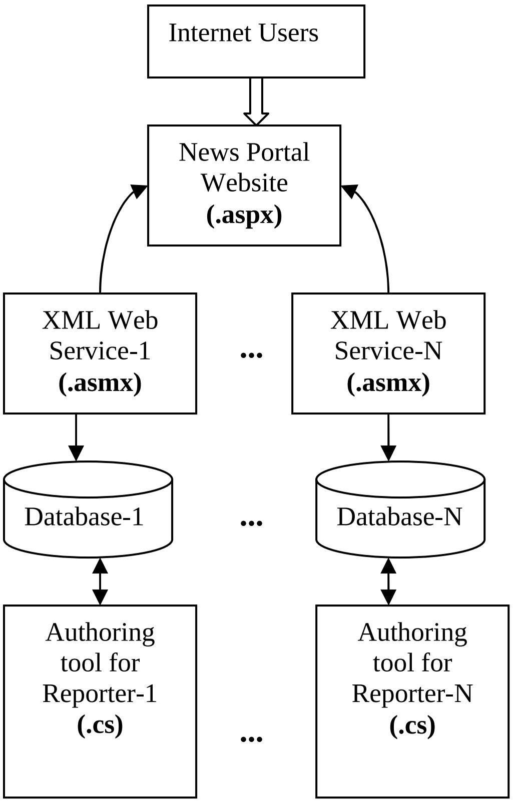
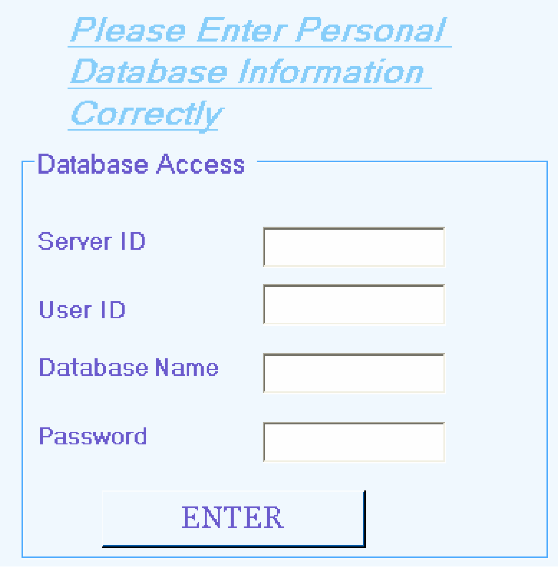

Internet Users
News Portal Website (.aspx)
XML Web Service-1 (.asmx)
...
XML Web Service-N (.asmx)
Database-1
...
Database-N
Authoring tool for Reporter-1 (.cs)
...
Authoring tool for Reporter-N (.cs)

Figure 1: The System Architecture

The *authoring tool* is a Windows client–side application implemented in the C# programming language on the .NET platform. It runs on the authors' computers and allows them to populate and maintain their own databases of articles on a database server accessible on the Internet. Authors can create new articles, edit existing ones, delete articles etc.

III. The "authoring tool" for reporters

The "authoring tool" is designed to demonstrate the viability of the proposed architecture, and is thus a bare-bones application, as are the other components of our implemented system. It consists mainly of two forms, named "Home" and "Reporter". Figure 2 depicts the "Home" form, which is presented to the user as soon as the program starts running. Basically, it gathers connection information to the database server from the user. The information entered on this form is then passed on to the "Reporter" form where it is used to actually make the necessary database connections.


Please Enter Personal Database Information Correctly

Database Access

Server ID

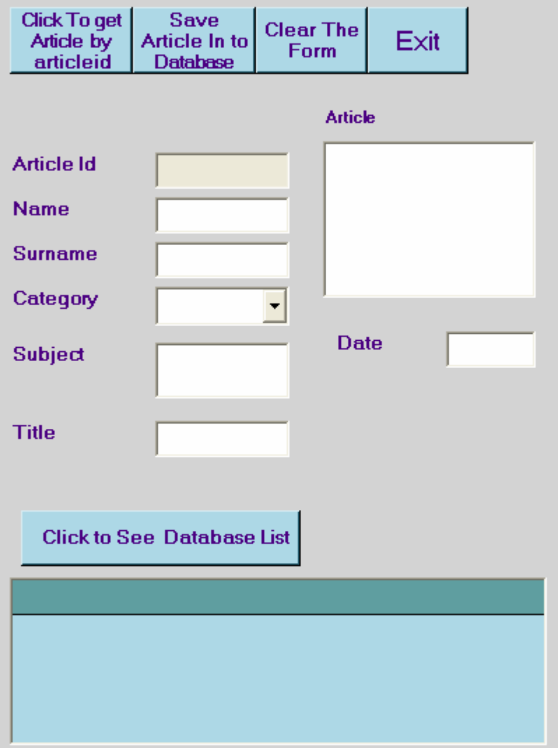
User ID

Database Name

Password

Figure 2: The "Home" form

Figure 3 presents the "Reporter" form. Using this form, an author (or *reporter* - we shall use these terms interchangeably in this paper) can save his articles to his database, can see his list of articles, can retrieve an article by its article identifier and clear the form.


Click To get Article by articleid
Save Article In to Database
Clear The Form
Exit

Article Id

Name

Surname

Category

Subject

Title

Article

Date

Click to See Database List

Figure 3: The "Reporter" form

```
SqlConnection con;
string constring = " Server= " + serverid + ";uid =
" + user + "; database = " + dbname +
";password=" + psswrd + " ";
con=new SqlConnection(constring);
con.Open();
string sql=" SELECT * FROM reporter
order by articleid desc ";
SqlDataAdapter da = new
        SqlDataAdapter(sql,con);
DataSet dst=new DataSet();
da.Fill(dst,"reporter");
dataGrid1.DataSource=dst.Tables[0];
con.Close();
```

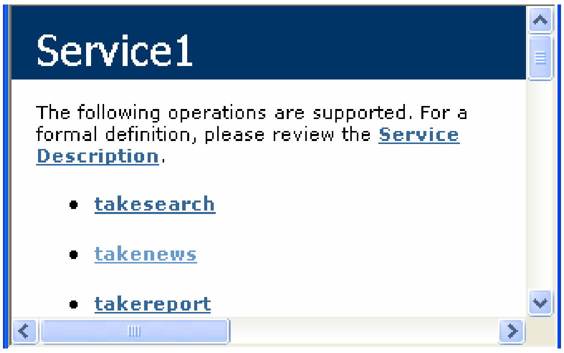
Figure 4: Code for establishing a database connection and listing the database contents

Just to give a taste of the type of code used for database operations, in Figure 4 we have the code for opening a connection to an SQL database server and retrieving the list of articles by an author. First, the code sets up a connection string and uses the string to open the connection. The variables “serverid”, “user”, “dbname” and “psswrd” are parameters from the “Home” form. Then a command string in SQL is written to retrieve articles from database in descending order. A “dataAdapter” object is used to query the database, and fill a “dataSet” object with the results of the query. The “dataSet” object is then bound to a “dataGrid” object, which displays the results in a visually attractive manner. We note in this code that the built-in functionality of objects made available to the programmer in the .NET framework resulted in very few lines of actual code being written by the programmer.

IV. The “article server” Web service

Figure 5 depicts the functions (so called “Web Methods” in .NET terminology) exposed by the Web service. The description of these functions are given below.

- **Takenews( ) WebMethod:** Returns the most recent article by an author.
- **Takereport(string no, string fname) WebMethod:** This function takes two parameters: an article ID (“no”), and an author name (“fname”). It returns the article belonging to the author, and which has the specified ID. This function has been made availabe for the case when, for some reason, two or more authors decide to use the same Web service to publish their articles.
- **Takesearch (string q) WebMethod:** This is the most general of the Web methods: it takes a query string (whose format should naturally be known to the calling program) containing complex search criteria, and returns all articles satisfying the criteria.



Service1

The following operations are supported. For a formal definition, please review the Service Description.

- takesearch
- takenews
- takereport

Figure 5: Web Service Functionality

To give a flavour of the programming required for programming Web services, the code of the “takereport” webmethod is given in Figure 6.

```
namespace news2
{
  public class Service1 :
   System.Web.Services.WebService
  {
     public Service1( ) {
      InitializeComponent( ); }

     public DataSet takereport(string no,string
   fname)   {
        con.Open();
        SqlDataAdapter DAnews = new
        SqlDataAdapter();
        SqlCommand report=new
        SqlCommand("select article from reporter
        where articleid= '"+Convert.ToInt32(no)+"'
        and name='"+fname+"'",con);
        DAnews.SelectCommand=report;
        DataSet dset = new DataSet();
        DAnews.Fill(dset, "reporter");
        return dset;
        con.Close(); }
  }
}
```

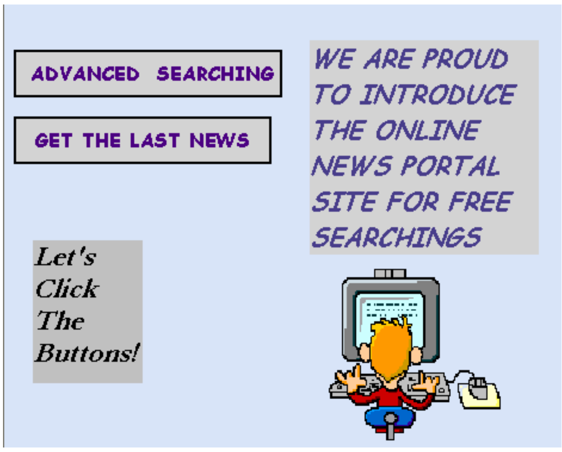
Figure 6: "Takereport" WebMethod Code

V. The “news portal” Web application

Our news portal Web application demonstrates how a news portal can use XML Web services to extract articles based upon a user’s choices. It is implemented as an ASP.NET application, which means an Internet browser application is required to use it.

An ASP.NET application consists of “Web forms”. Our portal has the Web forms ***index***, ***search***, ***detail*** and ***lastnews***. Their functionality is as follows:

1. **Index Web Form:** This is the main entry point into the portal, depicted in Figure 7. It presents a main menu about site content and navigation buttons.



WE ARE PROUD TO INTRODUCE THE ONLINE NEWS PORTAL SITE FOR FREE SEARCHINGS

Let's Click The Buttons!

Figure 7: Initial page of the news portal

From this page, by following the naviagation buttons on the page, a user can perform an advanced search for articles, or he may choose to get the latest news. In either case, he will be directed to a new Web form.

2. **Search Web Form:** On this form a user can initiate a detailed search for articles. Criteria can be specified for categories of articles, reporter name, subject, date, time interval etc. The results are shown in the a “datagrid” object on the screen. The user then can choose to view any article in detail by clicking on its name, and he will be transfered to the “Detail” Web form.

3. **Detail Web Form:** In this form, there is only a datagrid object which shows the full text of an article selected from “*search*” or “*lastnews*” pages.

4. **Lastnews Web Form:** This form displays the latest (most recent) articles of authors.

Our portal uses two Web services for its content. These are named **news** and **news2 in the application.** Figure 8 shows the code which invokes the “takesearch()” method of the two Web services.

```
dr =new DataSet();
drs =new DataSet();
news2 ser =new  Service1();
ser.Credentials =
System.Net.CredentialCache.DefaultCredentials
dr =ser.takesearch(q);
news servis=new Service1();
servis.Credentials =
System.Net.CredentialCache.DefaultCredentials;
drs=servis.takesearch(q);
```

Figure 8: Invoking methods of Web Services

VI. Related work

In [4] the authors present a domain-oriented approach to Web data extraction and discuss its application to automatically extracting news from Web sites. [5] describes a news recommendation system that allows a news service provider to analyze its customer profile and then produce customized news services. This is a new approach that builds customer profiles from their browsing behavior and analyzes these profiles to produce personal news delivered on the web. In [6] the author discusses the characteristics of portals defined as web-based interfaces to applications. Knowledge portals are explained in [7] as being intermediaries for knowledge access and knowledge sharing on the Web. Finally, machine learning technigues are advocated in [8] for the automatic creation and maintence of domain-specific Internet portals.

VII. Conclusion and future research directions

We described a scalable architecture for implementing news portals by making use of XML Web services. Our architecture achieves considerable scalability in terms of adding sources of content, flexibility of composing content, and decoupling between reporters and portals that make use of their articles. To demonstrate the feasibility of the idea, we implemented a proof-of-concept system in the .NET framework which highlights the main features of the proposed architecture.

Future work along these lines would be to incorporate authorization and authentication functionality into the Web services component, so that an author might selectively make his articles available to his customers (i.e. portals) and may be even charge them for retrieved articles.

REFERENCES

[1] http://www.portalscommunity.com/library/fundamentals.cfm

[2] http://news.google.com/

[3] http://news.yahoo.com/

[4] D. C. Reis, P. B. Golger, A.Silva and A. F. Lander, "*Automatic web news extraction using tree edit distance*", in: Proceedings of the 13th conference on the World Wide Web, 2004, New York, pp. 502-511.

[5] Hung–Jen Lai, Ting –Pen Liang and Y. C. Fu, "*Customized Internet news services based on customer profiles*", in: Proceedings of the 5th international conference on Electronic commerce, 2003, Pittsburgh , Pennsylvania, pp. 225-229.

[6] Geofrey G. Fox, "*Portals for Web Based Education and Computational Science*", ERDC Technical Report, May 2000.

[7] S. Staab and A. Maedche, "*Knowledge portals:ontologies at work*", AI Magazine, Vol.22, No.2, Summer 2001, pp. 63-75.

[8] A. K. McCallum, K. Nigam, J. Rennie, and Seymore, "*Automating the construction of internet portals with machine learning*", Information Retrieval, Vol.3, No.2, 2000, pp.127-163.