Web Service Based System for Generating Input Data Sets

Velin Kralev, Radoslava Kraleva

South-West University "Neofit Rilski", Blagoevgrad, Bulgaria

***Abstract**: This article deals with a three-layer architectural model of a distributed information system based on Web services, which will be used for automatic generation of sets of input data. The information system will be constructed of a client layer, a service layer and of a data layer. The web services as a tool of developing distributed software systems will be presented briefly. A web service and the implementation of its web methods will be described. A way to use the developed web methods in real application will be proposed.*

***Keywords:** web services.*

1. INTRODUCTION

When solving optimization problems typically different algorithms are used. The algorithms are compared on the basis of various criteria - performance, efficiency, etc. To do so it is necessary to have the same input data (benchmark) survey on which the algorithms to be applied. Preparation of such sets of input data, if done manually is extremely expensive. It is therefore necessary to use different methods and mechanisms through which input data (most often held in the form of vectors and matrices) to be automated.

It is precisely on this issue that this article focuses on. An approach for building and using a web service that will provide its consumers (applications) web methods to retrieve sets of input data will be proposed. These sets of input data can be generated in an automated way (i.e. their generation requires no user intervention). These data must be current, be obtained for reasonable time, and be able to be requested at a later stage in the same type.

2. WEB SERVICES AS A TOOL FOR DEVELOPMENT OF DISTRIBUTED SOFTWARE SYSTEMS

Currently, the development of server applications running on different operating systems has become a primary goal of developers [1]. Communication between application-application is not a new idea. There are already technologies like DCOM, RPC and others [5, 7] that allow this type of communication. The main limitation of these technologies is that they only work between two similar systems.

Web services represent a new technology designed to change the way of developing modern information systems. Communicate among different computers using web services is possible thanks to the protocol Simple Object Access Protocol (SOAP). It is built on the standard HTTP (Hypertext Transfer Protocol) protocol. Therefore, the Web server can handle SOAP requests and data packets can pass through firewalls. SOAP defines the XML-based record of making requests to the execution method of an object of the server, passing its parameters, and recording to define the shape of the response [3]. SOAP was originally designed to overcome the weaknesses of using DCOM (Distributed Component Object Model) in web servers. It is presented in the W3C for standardization and is approved [2]. SOAP replaced the calls by using a COM (Common Object Model) among different types of computers. Similarly, the definition of service SOAP format WSDL (Web Services Description Language) replaced IDL (Interface Description Language) standard libraries that are used by COM and COM+. WSDL documents are XML documents providing a definition of metadata for SOAP requests.

3. ARCHITECTURE SCHEME OF THE INFORMATION SYSTEM

In developing information systems based on Web services common functionality between different systems is shared. In other words, different systems use the same web methods that are available from a Web service.

This study was conducted to choose the model of information system that uses multi-layer architecture based on Web services. Other types of architectures of information systems (such as single-layer (monolithic), two-layer (client / server), etc.) restrict the number of layers that can be used. Thus the workload of the entire information system may not be distributed due to the inability of physical separation. We note that the two-layer architecture is only part of the functionality associated with the processing of data to be transferred on the server database.

The proposed architectural model of the system consists of three layers:

- Client layer. This layer includes thin clients (WinForms applications) and standard Web clients (browsers). It is built on client computers (PCs, laptops, etc.).
- Services layer. This layer provides functions as Web services. It is based on servers in the development of business objects, using rules for access to them. Also, queries are processed which form the responses to clients.

• Data layer. This layer actually consists of two sub-layers. One layer of data access which provides the necessary interfaces. The second layer is used for physical storage of the data. This layer offers the opportunity to work with data, such as execution of SQL constructions insert, update, delete and select.

The architectural scheme of information system using Web methods for retrieving sets of input data is presented at Fig 1.

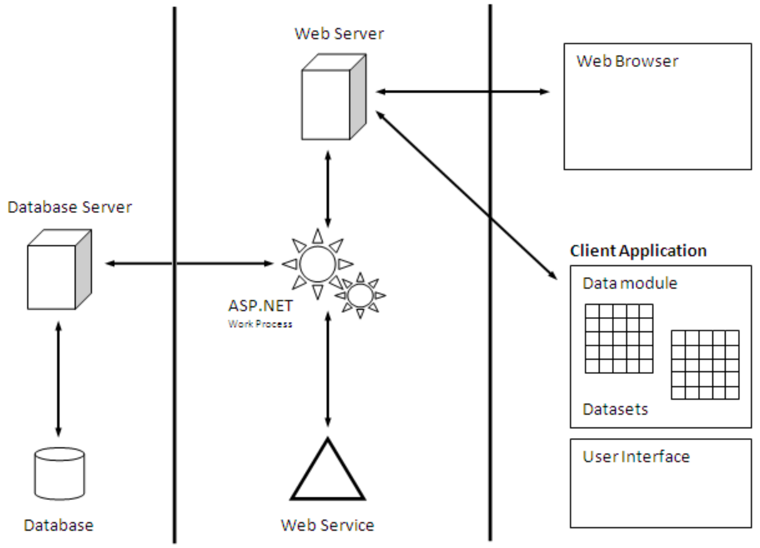


Fig 1: Architectural scheme of the information system.

The applications running in the client layer (Web forms or Win forms based) use SOAP protocol (transported by HTTP protocol) to communicate with the middle layer (i.e. service layer). The relationship between service layer and data layer is implemented by drivers for data access.

4. DEVELOPMENT OF WEB SERVICE FOR GENERATING INPUT DATA SETS

Here, the developed a Web service will be presented briefly. For its implementation the visual environment for design and event-oriented programming - Turbo Delphi for. NET is used. As a technology platform the Microsoft .NET Framework is used.

The web service will be associated with obtaining information from a relational database of existing information systems for automated generating of university timetable [5, 6]. Relational database management system is Microsoft SQL Server 2008.

The developed Web service provides two Web methods that are available to consumers:

1. The method GetSemestersInfoAsXML: String - this is a web method that sends back information in XML format for the semester, faculties, departments, majors and courses. This web method can be used by the Web service consumers to retrieve a list of identification numbers of courses (i.e. primary key values from the table of courses presented by the relational database in [6]). This list may be submitted as a parameter of the second web method.

The pseudo code of the first web method is presented at Fig 2.

```
function GetSemestersInfoAsXML: String;
var sXML: TStrings;
begin
 // created a connection to the database
 Connection := TADOConnection.Create;
//Setting the ConnectionString property and opening the connection
 //Dynamic creation of data sets: dsPeriods, dsFaculties,
 //dsDepartments, dsSpecialities and dsCourses
 // generating the resultant XML stream
 sXML := TStringList.Create;
 sXML.Add('<?xml version="1.0"');
 sXML.Add('<ttr:semesters xmlns:ttr='+
          '"http://timetable.swu.bg/TimetableResearch">');
 for each dsPeriods do begin
  sXML.Add('<ttr:semester name=' + dsPeriods.Name);
  sXML.Add(' id=' + dsPeriods.PrimaryKey + '>');
  for each dsFaculties do begin
   sXML.Add('<ttr:faculty name=' + dsFaculties.Name);
   sXML.Add('id=' + dsFaculties.PrimaryKey + '>');
   for each dsDepartments do begin
    sXML.Add('<ttr:department name=' + Departments.Name);
    sXML.Add('id=' + Departments.PrimaryKey + '>');
    for each dsSpecialities do begin
     sXML.Add('<ttr:speciality name=' + dsSpecialities.Name);
     sXML.Add('id=' + dsSpecialities.PrimaryKey + '>');
     for each dsCourses do begin
      sXML.Add('<ttr:course name=' + dsCourses.Name);
      sXML.Add('id=' + dsCourses.PrimaryKey + '/>');
     end; sXML.Add('</ttr:speciality>');
    end; sXML.Add('</ttr:department>');
   end; sXML.Add('</ttr:faculty>');
  end; sXML.Add('</ttr:semester>');
 end;  sXML.Add('</ttr:semesters>');
 Result := sXML;
end;
```

Fig 2: Pseudo code of a Web method GetSemestersInfoAsXML.

To generate the necessary information object of type TConnection is created through which to contact the database by setting the

ConnectionString property and calling the method Open. For each of the used tables the created object of the type DataSet, which is filled with information from the database. Then an object of type TStringList is created which is stored in XML format generated text. After the prologue of the XML file a namespace is declared, which is related with the developed Web service. The relationships between tables: semester, faculties, departments, majors and courses are one-to-many. Therefore, using nested loops that go any subset of related data, thus forming a hierarchical structure in which they organize their resources. At each step of each loop the corresponding XML node is constructed, adding the name and identification number of each resource. After each loop the corresponding XML block closes. After the outer loop the block of the main root element of the XML file is closed.

2. The method GetTimetableDataSetAsXML (ListOfCodeCourses: String): String - this is a web method on a list of numbers of courses (resulting from the first web method) sends back all relevant information resources within the participating timetable (courses, professors, auditoriums, events, matrices for the distribution of teachers, students and auditoriums in events, etc.).

The pseudo code of the second web method is presented at Fig 3.

```
function GetTimetableDataSet(ListCodeCourses: String): String;
var sXML: TStrings;
begin
 // created a connection to the database
 Connection := TADOConnection.Create;
 //Setting the ConnectionString property and opening the connection
 //Dynamic creation of data sets: dsLecturers, dsRooms, dsStudentsq,
 //dsEvents, dsKursInfo using ListCodeCourses
 // generating the resultant XML stream
 sXML := TStringList.Create;
 sXML.Add('<?xml version="1.0"');
 sXML.Add('<ttr:dataset xmlns:ttr='+
          '"http://timetable.swu.bg/TimetableResearch">');
 // adding a node for teachers
 sXML.Add('<ttr:lecturers count="' + dsLecturers.Count + '">');
 for each dsLecturers do
   sXML.Add('<ttr:lecturer name="' + dsLecturers.Name + '/>');
 sXML.Add('</ttr:lecturers>');
 // adding a node for auditoriums
 sXML.Add('<ttr:auditoriums count="' + dsRooms.Count + '">');
 for each dsRooms do
   sXML.Add('<ttr:auditorium name="' + dsRooms.Name + '/>');
 sXML.Add('</ttr:auditoriums>');
 // adding a node for students
 sXML.Add('<ttr:students count="' + dsStudents.Count + '">');
 for each dsStudents do
   sXML.Add('<ttr:students id="' + dsStudents.Id + '/>');
 sXML.Add('</ttr:students>');
 // adding a node for events
 sXML.Add('<ttr:events count="' + dsEvents.Count + '">';
 for each dsEvents do begin
  sXML.Add('<ttr:event id="' + /*and other attributes */ + '">');
  for each dsStudents do
   if dsStudents.Pk in dsEvents then /*if student in current event*/
     sXML.Add('<ttr:student id="' + dsStudentId + '"/>');
   sXML.Add('</ttr:event">';
  sXML.Add('</ttr:events">';
 end //for each dsEvents do
 sXML.Add('</ttr:dataset">';
end;
```

Fig. 3: Pseudo code of a Web method GetTimetableDataSetAsXML.

For the web method the connection to the database, dynamic creation of data sets, generating a list of strings and description of the main root element of the second XML file is similar to the first web method. Then alternately nodes for teachers, students and auditoriums are added. In the corresponding sub-nodes the available resources of each type are listed. A special case is the events where for each added event (in common node for the events) nodes for students attending any event are inserted. It is possible for a student node to be added as sub-nodes of more than one event.

Similarly as in the first web method, at the end of each loop the corresponding XML block closes. After the loop of events is closed blocks of the main root element of the XML file.

5. USE OF THE DEVELOPED WEB SERVICE

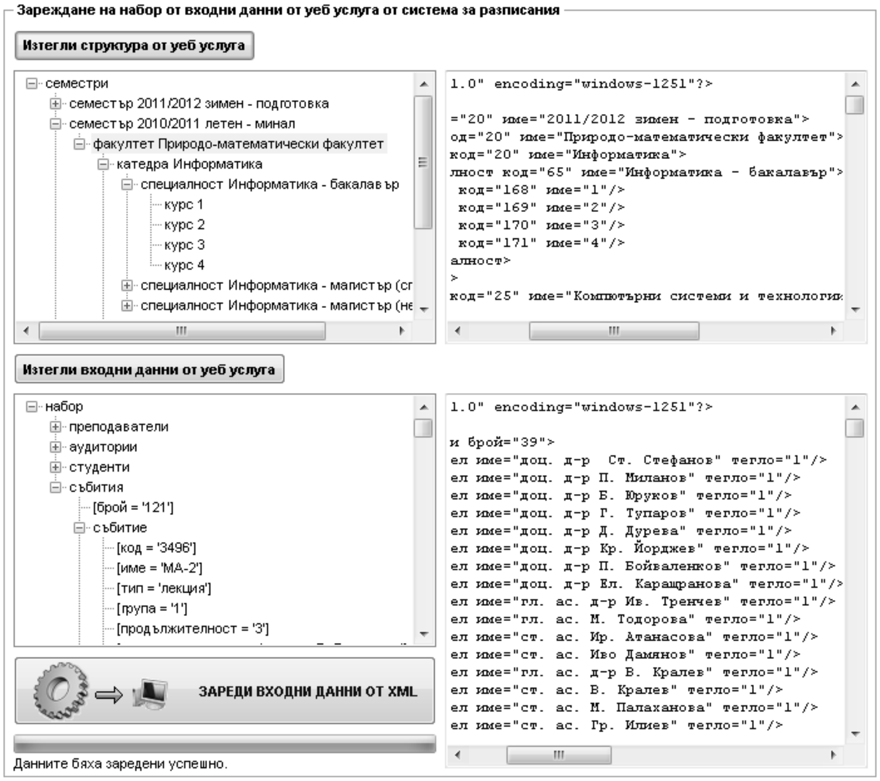
Each application must perform three steps in order to use a Web service. First - discovery of the service (this is the process of extracting information about a web service), second - generating the proxy class (a class that encapsulates the process of communication between the application and Web service) and third - using the proxy class to call methods web service [7].

We will modify the application that is used to study genetic and mimetic algorithms to solve optimization problems for automated generation of university timetables. The application is developed on Turbo Delphi for Win32 and is presented in [4]. Note that the web service can be used in applications developed with Java, Delphi 6, 7, or client built with. NET (e.g. Visual Studio for. NET).

To the the application under investigation a Web reference is added. The URL address to the WSDL document of the developed Web service is referred to. This WSDL document can be found at:

```
http://194.141.86.222/TimetableResearch/TimetableResearchWebService.asmx?wsdl
```

After adding a Web reference to the application, IDE automatically generates a proxy class that hides the specific means by calling the Web service methods. In other words, the proxy class is a class layer between the HTTP SOAP request to a web server and the code that is written for making this request [7]. In fact, this class works as any other class. Proxy class declares methods that can be called from the application consumed, which in turn invokes methods of a Web service itself.


Зареждане на набор от входни данни от уеб услуга от система за разписания
Изтегли структура от уеб услуга
Изтегли входни данни от уеб услуга
ЗАРЕДИ ВХОДНИ ДАННИ ОТ XML
Данните бяха заредени успешно.

```
семестри
    семестър 2011/2012 зимен - подготовка
    семестър 2010/2011 летен - минал
        факултет Природо-математически факултет
            катедра Информатика
                специалност Информатика - бакалавър
                    курс 1
                    курс 2
                    курс 3
                    курс 4
                специалност Информатика - магистър (сп
                специалност Информатика - магистър (не
```

```
1.0" encoding="windows-1251"?>

="20" име="2011/2012 зимен - подготовка">
од="20" име="Природо-математически факултет">
код="20" име="Информатика">
лност код="65" име="Информатика - бакалавър">
 код="168" име="1"/>
 код="169" име="2"/>
 код="170" име="3"/>
 код="171" име="4"/>
алност>
>
код="25" име="Компютърни системи и технологи
```

```
набор
    преподаватели
    аудитории
    студенти
    събития
        [брой = '121']
        събитие
            [код = '3496']
            [име = 'МА-2']
            [тип = 'лекция']
            [група = '1']
            [продължителност = '3']
```

```
1.0" encoding="windows-1251"?>

и брой="39">
ел име="доц. д-р  Ст. Стефанов" тегло="1"/>
ел име="доц. д-р П. Миланов" тегло="1"/>
ел име="доц. д-р Б. Юруков" тегло="1"/>
ел име="доц. д-р Г. Тупаров" тегло="1"/>
ел име="доц. д-р Д. Дурева" тегло="1"/>
ел име="доц. д-р Кр. Йорджев" тегло="1"/>
ел име="доц. д-р П. Бойваленков" тегло="1"/>
ел име="доц. д-р Ел. Каращранова" тегло="1"/>
ел име="гл. ас. д-р Ив. Тренчев" тегло="1"/>
ел име="гл. ас. М. Тодорова" тегло="1"/>
ел име="ст. ас. Ир. Атанасова" тегло="1"/>
ел име="ст. ас. Иво Дамянов" тегло="1"/>
ел име="гл. ас. д-р В. Кралев" тегло="1"/>
ел име="ст. ас. В. Кралев" тегло="1"/>
ел име="ст. ас. М. Палаханова" тегло="1"/>
ел име="ст. ас. Гр. Илиев" тегло="1"/>
```

Fig. 4: Sample session using the improved prototype.

An example of the work session with the improved prototype, which is described in [4] is shown at Fig 4. Additional functionality is added that uses the developed and presented in this paper Web service. The first method is used to retrieve information that is organized in a hierarchical form (semester, faculties, departments, majors, courses) regarding courses for which data are available that they have participated in the timetable for the selected semester.

The second web method used to derive data for the participating resources and their interconnections. The range of resources involved is determined by the given input parameter to the method which is a list of numbers of courses. Once the data have been received in XML format they complet the relevant vectors and matrices (according to the method used in the present prototype [4]).

6. CONCLUSION

In this article an architectural model of a distributed information system based on Web services for automated generation of input data sets is proposed. The web services as a tool for developing distributed software systems are presented briefly. A three-layer architecture scheme of an information system, which consists of a client layer, service layer and data layer is presented. A web service and the realization of the offered web methods are described. For both Web methods pseudo code for their development is presented. Web methods are implemented in a real application and are tested.

7. REFERENCES

[1] Bhasin, H. (2002) Microsoft ASP.NET Professional projects, Premier Press.

[2] Cantu, M. (2005) Mastering Borland Delphi 2005, Sybex.

[3] Kralev, V. (2006) The web services like an instrument of building the distributed software systems. Journal of the Technical University at Plovdiv "Fundamental Sciences and Applications", Vol. 13 (1), p.129-136.

[4] Kralev, V. (2009) A genetic and memetic algorithm for solving the university course timetable problem. International Journal on Information Technologies and Knowledge (IJ ITK), Vol.3.

[5] Kralev, V., Kraleva, R. (2009) Architecture model of information system for automated establishment of university course timetabling. Proceedings of the Second Balkan Scientific Conference The Science, the Education and the Art in 21st Century, Bulgaria, Blagoevgrad, 2008.

[6] Kralev, V., Kraleva, R., Siniagina, N. (2009) An integrated system for university course timetabling. Proceedings of the Third International Scientific Conference (FMNS2009), Blagoevgrad, Bulgaria, 2009. South-West University Publishing House.

[7] Pacheco, X. (2004) Delphi for .NET Developer's Guide, Sams Publishing.