IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
21-23 September 2009, Rende (Cosenza), Italy

# Interoperability of SCADA System Applications with Web Services

Arunas Lipnickas [1], Romas Rutkauskas [1], Ramunas Cerkauskas [2]
[1] Department of Control Technology, Kaunas University of Technology, Studentų str. 48-105, Kaunas, Lithuania;
e-mail: Arunas.Lipnickas@ktu.lt, Romas.Rutkauskas@ktu.lt
[2] Axis Industries, Kulautuvos g. 45a, Kaunas, Lithuania, e-mail: Ramunas.Cerkauskas@axis.lt

***Abstract - This paper describes the methodic of three-tier client-server architecture implementation for Web service creation. Data from SCADA system application are dynamically loaded to database by implementing ODBC method. For acceleration of data retrieving the parameters from SCADA application are transmitted to procedure that is stored in DB server. This stored procedure exports SCADA data to XML. DB interacts with Web server in XML. SOAP protocol is used for messaging with Web service client browser. The sequence of programming steps for interoperability with concrete Web service, which visualizes data, is shown.***

***Keywords - SCADA, automation, Web services, SOAP, XML***

## I. Introduction

New Web technologies and services directly influence industrial automation sphere. An Internet-based SCADA (Supervisory Control and Data Acquisition) systems become an important part of control system functioning. A human-machine interface or SCADA application can be integrated with a manufacturing execution system. For long time there were problem of communication between equipment with different incompatible data formants. The XML (Extensible Markup Language) with supporting family of technologies offered new possibilities for incompatibility problem solving. Now it is possible to track availability or status of a particular PLC (Programmable Logical Controller) and monitor data of technological processes remotely on browser via the request/response of Web services (WS).

By W3C [1] definition Web service is a software system designed to support machine-to-machine interaction over a network. Public interfaces and bindings of WS are defined and described using XML. WS can be accessed over the Web or any network from another application using RPC (Remote Procedure Call) style calls encoded using SOAP (Simple Object Access Protocol) [2]. Systems interact with the Web service in a manner prescribed by its interface description using WSDL (Web Service Description Language) [3]. SOAP messages typically are conveyed using HTTP (HyperText Transfer Protocol) with an XML in conjunction with other Web-related standards.

This article describes designed system and steps of Web service creation. It shows availability and importance of XML for SCADA system data monitoring and visualization.

## II. Situation in Industrial Automation Sphere

Software is an integral part of modern control systems. Automation architectures must provide consumers with primary services: control, configuration, collection and visualization of increasingly complex data sets. Enterprise networks are dynamic, continually evolving to meet the demands of end users and applications. Each step in this evolution is the result of challenges, including increased bandwidth demands and convergence to IP on factory floor [4, 7]. No matter how data are obtained, they in many implementations must be visualized in handy form for consumers. The need for remote intranet or internet access to industrial process information relatively reduces the need of local visualization on HMI (Human-Machine Interface) stands and increases the role of data visualization in browsers or other mobile equipment.

Information technologies merge with automation and control systems. This process affects industrial automation as well as all related infrastructure: programming, protocols and services, new kinds of transmission media, smart wireless transducers and actuators that are in locally or globally distributed areas and can be controlled by wired and wireless media or through the Internet.

Client-server architecture is becoming a typical solution for data exchanging and processing. Many of distributed control systems have complex data sets and structures and different data formats (especially from different vendors). Earlier data exchanging between various fieldbuses (devices) was complicated or even impossible. Now this problem can be solved by means of an XML. XML allows exchanging of different formats data and guarantees interoperability between devices connected to a digital fieldbus [5, 7].

The convergence of technologies and digitization of appliances influence the demands for industry automation. The network enables the distribution of data between different consumer devices. The Internet serves as a platform that enables an entirely new generation of consumer applications: remote and distributed control or monitoring and SCADA data visualization. That means that consumers now have a standardized interface for reading or writing of information that dynamically changes anywhere within the operational domain.

978-1-4244-4881-4/09/$25.00 ©2009 IEEE

## III. WEB SERVICES PROTOCOL STACK AND AUTOMATION NEEDS

Data acquisition and control techniques include many areas of data visualization and data pre-processing (fusion, editing, transformation, filtering, sampling, etc.). A Web service supports direct interactions with other software applications using XML based messages via Internet-based protocols. Web service is platform and language-independent, because it uses XML [7].

Web services allow easy obtain information from different locations and systems without requiring the end-user to learn how to connect to the other systems. Web services protocol stack consists of 4 layers. Main parts of the Web services architecture are shown in Fig.1.

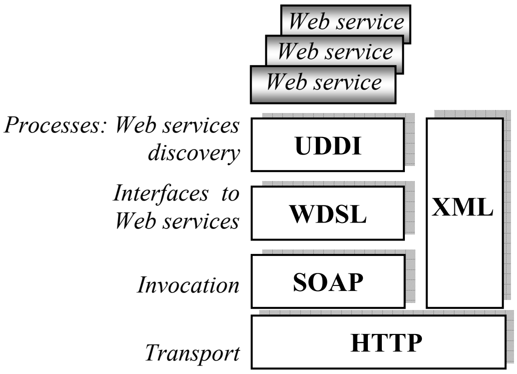


Fig.1. Typical Web service architecture.

The traditional interface model for a Web application: the user requests the server for a data, which are delivered to the browser. The first component of WS architecture is service discovery that allows locating one particular service from a collection of possible Web services. Interfaces to Web services are created in WSDL. Services are self-describing, because they announce how to invoke operations that they support.

Invocation allows passing messages between the server and the client. SOAP specifies format of requests and responses to/from the server. Practically SOAP is used in most cases of service invocation. The most popular protocol used for transportation is HTTP because firewalls don't perturb HTTP transactions.

WSDL describes a Web services as a collection of ports and operations. WSDL specification provides an XML format for documents and describes the public interface to the Web service. WSDL specifies interface, network address, data fields and types, message structure and bindings.

SOAP protocol supports structured data. Protocol is transport, encoding, programming language and platform independent. Helpfulness of SOAP for control and automation needs brings out in exchanging of structured and typed information between peers in a decentralized, distributed environment. SOAP message is formally specified as XML information set, which provides structured and abstract description of its contents.

SOAP is fundamentally a stateless, one-way message exchange paradigm, but applications can create more complex interaction patterns (e.g., request/response, request/multiple responses, etc.) by combining such one-way exchanges with features provided by an underlying protocol and/or application-specific information. SOAP provides the framework by which application-specific information may be conveyed in an extensible manner.

Web services are one of the core elements of the new Microsoft .NET Framework strategy. The idea is that remote applications will use XML to communicate with central systems, leaving the developer free to write in the language that is best suited for the job in hand. The .NET Framework is Microsoft's managed code programming model for building applications of many types: Web applications, server applications, smart client applications, console applications, database applications, and more.

## IV. STRUCTURE OF DESIGNED SYSTEM

The aim of designed system (Fig. 2) is to visualize data from PLC's that are connected via Ethernet network. The SCADA system periodically polls PLCs and writes data to database through ODBC (Open Database Connectivity) interface.

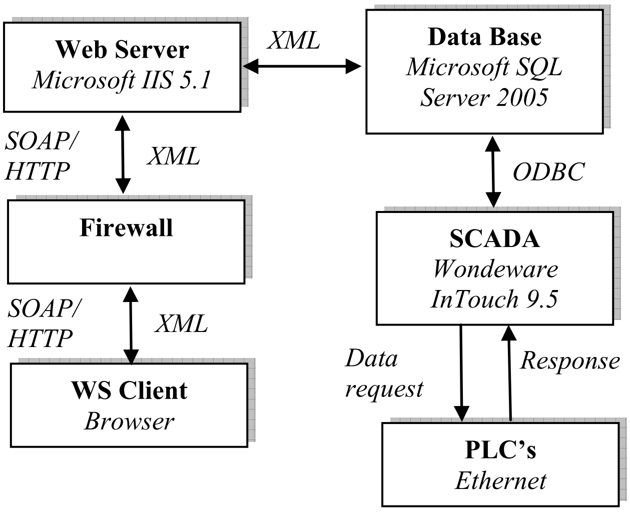


Fig. 2. Structure of PLC data visualization system.

Web Server communicates with DB Server in XML format, storing requested data in XML. SCADA system uses supervisory control and data acquisition software Wonderware InTouch 9.5. It provides quick and easy way to create human-machine interface (HMI) applications for Microsoft Windows operating systems. Wonderware Intouch 9.5 can function as a client and a server for both DDE (Dynamic Data Exchange) and SuiteLink communication protocols. It can be connected to industrial I/O systems and other Microsoft Windows applications.

SCADA periodically polls PLCs and, using stored procedures, writes process data to Microsoft SQL Server 2005 database. SQL (Structured Query Language) is used for searching, writing and deleting data from DB. For faster data manipulation process, saving and execution of SQL code is performed using stored procedures. Execution of stored procedures on server side is much faster than using SQL commands for data manipulation on client side. Executing, testing and optimization of stored procedures is independent from other applications.

Stored procedures are written in SQL language using Microsoft SQL server 2005 software. They are important for client-server database systems, because storing on the server side means, that procedure is available to all clients. When the stored procedure is modified, all clients automatically get the new version of code.

The SCADA system interacts with DB using Microsoft ODBC (Open Database Connectivity) method which allows the applications to access data from a various database management systems. SQL Server driver is used for data queries of SCADA application's translation into commands that DBMS (Data Base Management System) understands.

## V. SCADA Application and Web Service Programming

Here we will show the concrete example of Web service creation for automated cheese weighing process. Application and Web service creation steps can be as follows (Fig. 3):

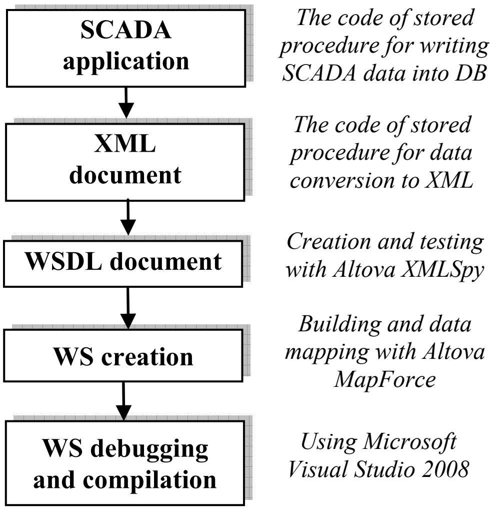

SCADA application
The code of stored procedure for writing SCADA data into DB
XML document
The code of stored procedure for data conversion to XML
WSDL document
Creation and testing with Altova XMLSpy
WS creation
Building and data mapping with Altova MapForce
WS debugging and compilation
Using Microsoft Visual Studio 2008

Fig. 3. Web service creation steps.

The code of stored procedure for SCADA data writing into DB is shown in Fig. 4. The execution of procedure in DB Server is initiated by itself SCADA system. Statement *Insert* is used to write data to DB, statement *Update* – renew old data.

Data, that are written into the database, are also shown in SCADA control and monitoring windows. Data representation is initiated by stored procedure "EXEC dbo.spRC_BizerbaSVSelect" with parameter "@SVGSverimoNr".

For DB data conversion to XML format another stored procedure (Fig. 5) is used. This procedure, accordingly to SCADA system specified criterions, retrieves appropriate data and represents them in XML format. For data retrieval statement *Select* is used. Function FOR XML is used for data representation in XML format. Represented in XML format data form a root element named <SverimoVaztarastis>.

Root element of XML document is specified within FOR XML function parameters. Web Server is used for storing data in XML format. Execution of stored procedure in DB Server is initiated by SCADA system.

```
-- Declaration of variables
@DezNrBzrb INT,@NrIsBizrb INT,@Brutto FLOAT,
@Tara FLOAT,@Netto FLOAT,@PadNr INT,@SVNr
NVARCHAR(50),@GamData NVARCHAR(50),
@RD_ID NVARCHAR(50)
AS
BEGIN TRAN
-- Data writing from SCADA system to DB
              INSERT INTO dbo.tbl_SVGBizerbaSV
(SverimoNr,Padeklas,DezNr,Numeris,Kiekis,Brut
to,Tara,Netto,RDID,GamybosData,DataLaikas)
        VALUES
(@SVNr,@PadNr,@DezNrBzrb,@NrIsBizrb,1,@Brutto
,@Tara,@Netto,@RD_ID,@GamData,
        ((convert(nvarchar(50), getdate(),
102))+' '+(convert(nvarchar(50), getdate(),
108))))
-- Assignment of values to stored procedure
variables
        SET @KiekisSUM = (SELECT SUM(Kiekis)
from dbo.tbl_SVGBizerbaSV WHERE Padeklas =
@PadNr and SverimoNr = @SVNr);
        SET @BruttoSUM = (SELECT SUM(Brutto)
from dbo.tbl_SVGBizerbaSV WHERE Padeklas =
@PadNr and SverimoNr = @SVNr);
        SET @NettoSUM = (SELECT SUM(Netto)
from dbo.tbl_SVGBizerbaSV WHERE Padeklas =
@PadNr and SverimoNr = @SVNr);
        SET @TaraSUM = (SELECT SUM(Tara) from
dbo.tbl_SVGBizerbaSV WHERE Padeklas = @PadNr
and SverimoNr = @SVNr);
--Data renewal in the database
        UPDATE dbo.tbl_SVGBizerbaSVSUM SET
Kiekis = @KiekisSUM ,
        Brutto = @BruttoSUM , Tara = @TaraSUM
, Netto = @NettoSUM
        where Padeklas = CAST (@PadNr AS INT)
AND SverimoNr = @SVNr
--Data, written in the database, are also
shown in the SCADA system
EXEC dbo.spRC_BizerbaSVSelect @SVGSverimoNr =
@SVNr
COMMIT TRAN
```

Fig. 4. Stored procedure for SCADA data writing into DB.

Web service in Microsoft IIS 5.1 server was created using graphical data mapping, conversion and integration tool Altova MapForce 2009 [6]. This data mapping application maps between any combinations of XML, database and Web service, then instantly transforms data or auto generates royalty-free data integration code for the execution of recurrent conversions.

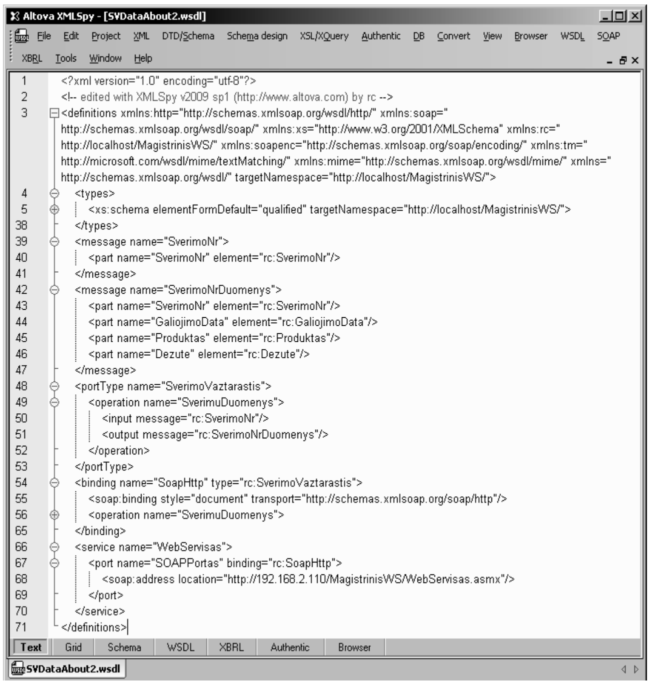
Using Altova XMLSpy editor, WSDL file was created (Fig. 6). This file describes Web service: what protocol is used, port type, location on the network and possible operation, that service can to perform. In this file document style, used for data exchange, is also defined, because Web service and browser uses data in XML format. Web service interacts with internet browser using request-response operation mode, while for data representation and request transferring is used SOAP protocol.

```
-- Declaration of variables
@SVGSverimoNr INT,@SurioKodasAxaptai INT
AS
BEGIN TRAN
set @SVGSverimoNrM = @SVGSverimoNr - 8 set
@len = len(@SVGSverimoNrM)
-- Verification of Wonderware InTouch
parameters transferred to stored procedure
while @len < 4
      begin
       set @SVGSverimoNrM =
'0'+@SVGSverimoNrM
       if len(@SVGSverimoNrM) = 4
        break
        else
        continue
      end
set @yy = left(@SVGSverimoNrM,2) set @svnr =
right(@SVGSverimoNrM,4)
set @SVGSverimoNrM = '08'+'-'+cast(@svnr as
nvarchar(4))
--Data retrieval and representation in XML
format
SELECT TOP 1 SV.SverimoNr as "@SverimoNr." ,
Uzsak.GaliojimoData as "@GaliojimoData",
'Riebus, kietas fermentinis sūris 50 % rieb.'
as "@Produktas",(
              SELECT TOP 4 SV.DezNr AS
"@Nr.", SV.GamybosData AS "GamybosData",
SV.RDID as "RDID", SV.Kiekis AS "Kiekis",
cast(round(SV.Brutto,3) as numeric(4,3)) as
"Bruto",cast(round(SV.Tara,3) as
numeric(4,1)) as "Tara",
cast(round(SV.Netto,3) as numeric(4,3)) as
"Netto" FROM dbo.tbl_SVGBizerbaSV as SV INNER
JOIN dbo.tbl_RealizUzsakymaiSVG as Uzsak ON
SV.SverimoNr = Uzsak.SverimoNr
WHERE SV.SverimoNr = @SVGSverimoNrM FOR XML
PATH('Dezutes'),TYPE
                 )
FROM dbo.tbl_SVGBizerbaSV as SV INNER JOIN
dbo.tbl_RealizUzsakymaiSVG as Uzsak ON
SV.SverimoNr = Uzsak.SverimoNr
WHERE SV.SverimoNr = @SVGSverimoNrM
FOR XML
PATH('NaujasSverimas'),ROOT('SverimoVaztarast
is'),TYPE ) SELECT @XML
```

Fig. 5. Stored procedure for data conversion to XML.

Web service data mapping structure (Fig. 7) consists of three components: data input, data output and data reading from XML file. XML file is located in Web Server. Accordingly to requested parameters, data input component performs general data assignment, data output component performs general data and parameters values from XML file.

Data from Web service to intranet or intranet client's browsers are transferred in XML format over SOAP protocol. Requests of clients and Server responses use HTTP protocol that passes through existing firewall.


Altova XMLSpy - [SVDataAbout2.wsdl]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser WSDL SOAP
XBRL Tools Window Help

```
<?xml version="1.0" encoding="utf-8"?>
<!-- edited with XMLSpy v2009 sp1 (http://www.altova.com) by rc -->
<definitions xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:soap="
http://schemas.xmlsoap.org/wsdl/soap/" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:rc="
http://localhost/MagistrinisWS/" xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/" xmlns:tm="
http://microsoft.com/wsdl/mime/textMatching/" xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns="
http://schemas.xmlsoap.org/wsdl/" targetNamespace="http://localhost/MagistrinisWS/">
  <types>
    <xs:schema elementFormDefault="qualified" targetNamespace="http://localhost/MagistrinisWS/">
  </types>
  <message name="SverimoNr">
    <part name="SverimoNr" element="rc:SverimoNr"/>
  </message>
  <message name="SverimoNrDuomenys">
    <part name="SverimoNr" element="rc:SverimoNr"/>
    <part name="GaliojimoData" element="rc:GaliojimoData"/>
    <part name="Produktas" element="rc:Produktas"/>
    <part name="Dezute" element="rc:Dezute"/>
  </message>
  <portType name="SverimoVaztarastis">
    <operation name="SverimuDuomenys">
      <input message="rc:SverimoNr"/>
      <output message="rc:SverimoNrDuomenys"/>
    </operation>
  </portType>
  <binding name="SoapHttp" type="rc:SverimoVaztarastis">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="SverimuDuomenys">
  </binding>
  <service name="WebServisas">
    <port name="SOAPPortas" binding="rc:SoapHttp">
      <soap:address location="http://192.168.2.110/MagistrinisWS/WebServisas.asmx"/>
    </port>
  </service>
</definitions>
```

Fig. 6. Web service description in WSDL document.

Web service code in C# language was generated using Altova MapForce software, debugged and compiled using Microsoft Visual Studio 2008 programming software. For Web service accessing from the intranet or internet, virtual directory, which stores generated source code, was created in Web Server.

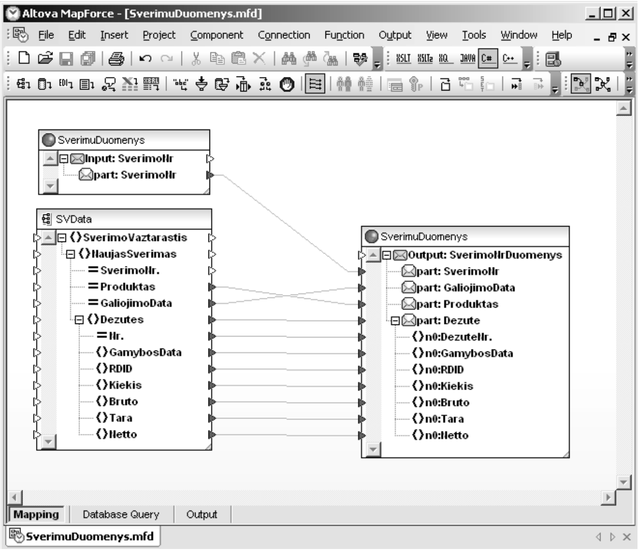


Fig. 7. Web service data mapping with Altova MapForce.

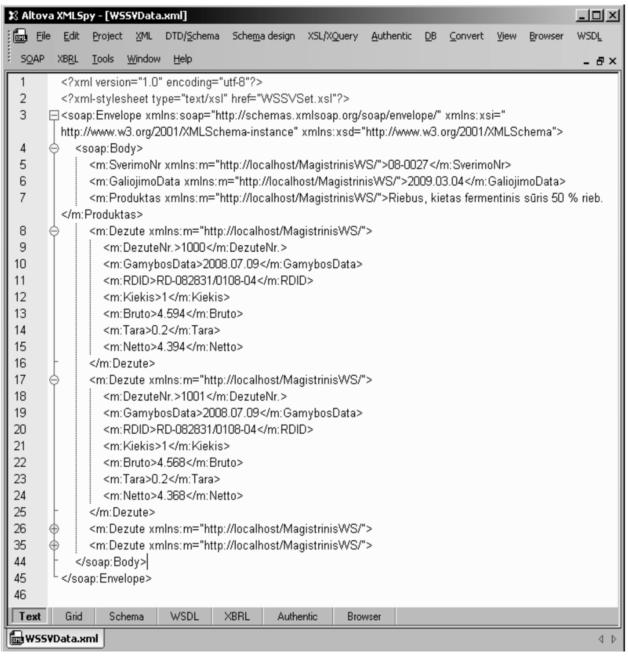
Using Altova XMLSpy editor, accordingly to WSDL file structure, SOAP request (Fig. 8) was created, which was used to test functionality of Web service.

For representation of SCADA system data in internet browser SOAP request was sent to Web server using address: „http://192.168.2.110/MagistrinisWS/WebServisas.asmx“.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:SOAP-
ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
          <SOAP-ENV:Body>
                    <m:SverimoNr
xmlns:m="http://localhost/MagistrinisWS/">08-0027</m:SverimoNr>
          </SOAP-ENV:Body>
```

Fig. 8. SOAP request in XML format.

Web service response in XML format is shown in Fig. 9.


Altova XMLSpy - [WSSVData.xml]
File Edit Project XML DTD/Schema Schema design XSL/XQuery Authentic DB Convert View Browser WSDL
SOAP XBRL Tools Window Help
Text Grid Schema WSDL XBRL Authentic Browser
WSSVData.xml

```
<?xml version="1.0" encoding="utf-8"?>
<?xml-stylesheet type="text/xsl" href="WSSVSet.xsl"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="
http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <m:SverimoNr xmlns:m="http://localhost/MagistrinisWS/">08-0027</m:SverimoNr>
    <m:GaliojimoData xmlns:m="http://localhost/MagistrinisWS/">2009.03.04</m:GaliojimoData>
    <m:Produktas xmlns:m="http://localhost/MagistrinisWS/">Riebus, kietas fermentinis sūris 50 % rieb.
</m:Produktas>
    <m:Dezute xmlns:m="http://localhost/MagistrinisWS/">
      <m:DezuteNr.>1000</m:DezuteNr.>
      <m:GamybosData>2008.07.09</m:GamybosData>
      <m:RDID>RD-082831/0108-04</m:RDID>
      <m:Kiekis>1</m:Kiekis>
      <m:Bruto>4.594</m:Bruto>
      <m:Tara>0.2</m:Tara>
      <m:Netto>4.394</m:Netto>
    </m:Dezute>
    <m:Dezute xmlns:m="http://localhost/MagistrinisWS/">
      <m:DezuteNr.>1001</m:DezuteNr.>
      <m:GamybosData>2008.07.09</m:GamybosData>
      <m:RDID>RD-082831/0108-04</m:RDID>
      <m:Kiekis>1</m:Kiekis>
      <m:Bruto>4.568</m:Bruto>
      <m:Tara>0.2</m:Tara>
      <m:Netto>4.368</m:Netto>
    </m:Dezute>
    <m:Dezute xmlns:m="http://localhost/MagistrinisWS/">
    <m:Dezute xmlns:m="http://localhost/MagistrinisWS/">
  </soap:Body>
</soap:Envelope>
```

Fig. 9. Web service response in XML format.

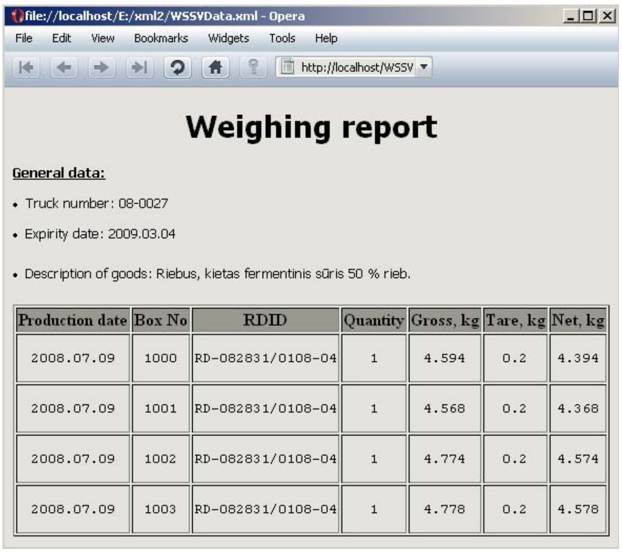
Web Server communicates with DB Server in XML format, storing requested data in XML. There are two major ways to display XML in a Web browser:

The first way is to translate XML into HTML – usually with a scripting program (JavaScript, VBScript).

The second way is to use XSL (Extensible Stylesheet Language) to define the output format of the XML.

Style sheet describes rules for presenting of XML documents. Once in XML format, objects can be sent through standard Internet communication channels and be addressed to any connected device.

Weighing report XML data in Opera 10 internet browser (Fig. 10) are displayed using XSL (Extensible StyleSheet Language), which generates output format in HTML from data that are in XML format. Thanks to HTML, XML data can be viewed in all HTML format supporting internet browsers.


file://localhost/E:/xml2/WSSVData.xml - Opera
File Edit View Bookmarks Widgets Tools Help
http://localhost/WSSV

**Weighing report**

**General data:**

- Truck number: 08-0027
- Expirity date: 2009.03.04
- Description of goods: Riebus, kietas fermentinis sūris 50 % rieb.

| Production date | Box No | RDID | Quantity | Gross, kg | Tare, kg | Net, kg |
|---|---|---|---|---|---|---|
| 2008.07.09 | 1000 | RD-082831/0108-04 | 1 | 4.594 | 0.2 | 4.394 |
| 2008.07.09 | 1001 | RD-082831/0108-04 | 1 | 4.568 | 0.2 | 4.368 |
| 2008.07.09 | 1002 | RD-082831/0108-04 | 1 | 4.774 | 0.2 | 4.574 |
| 2008.07.09 | 1003 | RD-082831/0108-04 | 1 | 4.778 | 0.2 | 4.578 |

Fig. 10. Weighting report data in Opera 10 browser.

## VI. Conclusion

In this paper we have shown how SCADA system benefits from an XML and Web services for achieving objectives of integration. The methodic of three-tier client-server architecture implementation for Web service creation is described. Data from SCADA system application are dynamically loaded to database by implementing ODBC method. The parameters from SCADA application for acceleration of data retrieving are transmitted to special procedure that is stored in DB server. This stored procedure exports SCADA data to XML format. DB interacts with Web server in XML. Internet-based SCADA is now a reality and offers many benefits. The usefulness of XML and Web services technologies for process automation field was shown in designed application.

## References

[1] Web Services Architecture. [Interactive]. [Accessed 15 June 2009], Available from <http://www.w3.org/TR/ws-arch/>.

[2] SOAP Version 1.2 [interactive]. Part 1: Messaging Framework (Second Edition) [accessed 15 June 2009]. Available from <*http://www.w3.org/TR/soap12-part1/*>.

[3] Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. W3C Recommendation 26 June 2007. [Interactive], [accessed 15 May 2009] Available from <*http://www.w3.org/TR/2007/REC-wsdl20-20070626/*>.

[4] A. Lipnickas, R. Rutkauskas, "New challenges in education of control technologies," 14th International Congress of Cybernetics and Systems of WOSC, September 9-12, 2008, Wroclaw, Poland, Wroclaw University of Technology, 2008. pp. 857-866.

[5] R. Rutkauskas, A. Lipnickas, C. Ramonas, V. Kubilius, "New Challenges for Interoperability of Control Systems," Electronics and Electrical Engineering, no. 3(83), 2008, pp. 71-74.

[6] MapForce – graphical data mapping, conversion, and integration tool. [Interactive], [accessed 15 January 2009] Available from <http://www.altova.com/products/mapforce/data_mapping.html.

[7] R. Rutkauskas, V. Mačerauskas, *Networks for building and industry automation*. - Kaunas: Technologija, 2004, 385 p.