605. Formalization of the observations of the sea level variations using XML Data schemas and Scalable Vector Graphics format

E. Paršeliūnas, L. Marozas, P. Petroškevičius, A. Zakarevičius, J. Stankūnas, V. Č. Aksamitauskas, A. H. Marcinkevičius
Vilnius Gediminas Technical University,
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania
e-mail: *gi@vgtu.lt*
Phone: +37052744703; **Fax:** +37052744705

(Received 11 November 2010; accepted 9 December 2010)

Abstract. XML (Extended Markup Language) Data Schemas as the format for information exchange in graphical SVG (Scalable Vector Graphics) format is presented in the paper. SVG file format is a language for two-dimensional graphics and is based on XML. It is suggested to apply it as the advanced version of previous data exchange format TGINEX (Tide Gauge Independent Exchange Format), which contains the plain ASCII data. SVG graphics are formatted from the observations of sea level observations by the tide gauges. As an example, the formalization of data is shown on the base of the sea level observing station KLPD, which is a part of the European Sea Level Service network.

XML Data Schemas, in addition, adds header’s information about the marine measurements' site, sensors and any additional necessary information. The correct header formation and the advantages of such data exchange format are analysed. Header is important for exchange between different sites from different countries as so it would be information standartization. Visual appearance of the SVG file, the source, headers and formation of the file in script are described in this paper also.

Keywords: sea level variations, Data Schemas, XML, SVG, TGINEX, sea level observations.

Introduction

Data exchange for marine research sites is extremely important and valuable [1–7]. At present TGINEX format (only ASCII data) is used for sea level observations data exchange (Fig. 1) [8–9].

```
BEGIN OF HEADER
Format                          : TGINEX version 1.0
PSMSL code                      : 080161
ID                              : KLPD
Station name                    : Klaipeda
Country                         : Lithuania
Contributor                     : Marine Research Center; VGTU GI
Latitude                        :  55.7165
Longitude                       :  21.1183
Datum                           : Normaal Amsterdams Peils(NAP)
Instrument type                 : Float
Instrument precision (mm)       :  10
Start Date and time (UTC)       : 2009/12/01 00:00:00
End Date and time (UTC)         : 2009/12/31 23:00:00
Record interval (in seconds)    : 60
Sampling interval (in minutes)  : 60
```

```
Quality control level           : L1
Parameter 1                     : Date (yyyy/mm/dd)
Parameter 2                     : Time (hh:mm:ss)
Parameter 3                     : QC flag
Parameter 4                     : Observed sea level (m)
Parameter 5                     : QC flag
END OF HEADER
2009/12/01 00:00:00 1 +0.2600 1
2009/12/01 01:00:00 1 +0.2600 1
2009/12/01 02:00:00 1 +0.2400 1
2009/12/01 03:00:00 1 +0.2300 1
2009/12/01 04:00:00 1 +0.2200 1
2009/12/01 05:00:00 1 +0.1900 1
2009/12/01 06:00:00 1 +0.1800 1
<...>
```

Fig. 1. Excerpt from the TGINEX file of KLPD station data

Since CSV (comma separated values) or SDF (standard data format) formats save place and as raw textual data can be exchanged, interpreted and imported easily, therefore graphical representation of data remains complicated. After TGX format was implemented, next short step was introduction of CGM format for marine data exchange [9]. However because of some disadvantages, the complete level ahead of TGINEX data is though now currently under development, but already functioning SVG (Scalable Vector Graphics) file format with XML Data Schemas [10–13].

Structure of SVG file

The SVG file is composed from two main parts: XML Data Schemas and observation data itself. The XML Data Schemas describe the structure of the marine research site: information on observing agency, equipment, sensors, data formats etc. First of all, the header of the marine sea level measurements' site is presented in XML Data Schemas. Excerpt from the header from KLPD site is presented in Fig. 2.

```
     <...>
     <xsd:element                                name="formInformation"
type="mi:formInformationType"/>
     <xsd:element                             name="siteIdentification"
type="mi:siteIdentificationType"/>
     <xsd:element name="siteLocation" type="mi:siteLocationType"/>
     <...>
     <xsd:element                               name="waterVaporSensor"
type="equip:waterVaporSensorType"/>
     <xsd:element                           name="otherInstrumentation"
type="equip:otherInstrumentationType"/>
     <...>
     <xsd:element                                  name="contactAgency"
type="contact:contactInformationType"/>
     <xsd:element                              name="responsibleAgency"
type="contact:contactInformationType"/>
     <xsd:element                                name="moreInformation"
type="mi:moreInformationType"/>
     <...>
```

Fig. 2. Excerpt from the XML Data Schemas header

XML Data Schemas can be easily extended or narrowed by the wish of responsible people. Currently there is work in progress on form with the most common information – so as XML Data Schema would be easily modified without actually having to reprogram anything.

Using of XML Data Schemas gives some advantages for data exchange purposes [14]. Firstly, it's great strength is that XML Data Schemas are written in XML, so it becomes easier to parse data with simple XML parser, then comes the security aspect of XML Data Schemas, extensibility, readability etc. Secondly, XML gives an opportunity to use SVG as vector graphics format - very powerful technology to visualize the technological data without loosing ability to check or parse raw textual data. For example, graphical representation of sea level hourly means data from KLPD site is presented in Fig. 3 (hourly file contains the means of the sea level observations at each minute; observations are recorded at each 10 seconds).

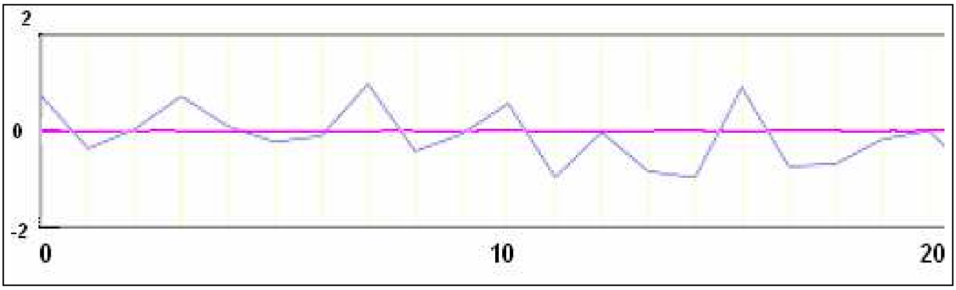

Fig. 3. Sea level hourly means data graphical representation from SVG file (first 20 minutes)

Excerpt from the simple source code of SVG file, representing the results of sea level observations, is given in Fig. 4.

```
<?xml version="1.0" standalone="yes"?>
<!DOCTYPE svg PUBLIC "" "" [
<!ENTITY sto0 "fill:#CCF2FF; stroke:none; stroke-width:0.028">
<!ENTITY stt2 "font-family:Arial; font-size:0.805; fill:#4B4B4B"> ]>
<svg xml:space="preserve" width="300mm" height="150mm" viewBox="00 -
1.50 12 1.50">
<g fill="none" > </g>
%Header

<EQUIPMENT>
<klpd.Format>TGINEX version 1.0</klpd.Format>
<klpd.PSMSLCode>080161</klpd.PSMSLCode>
<klpd.BaseEquipmentLib>BaseEquipmentLib</klpd.BaseEquipmentLib>
<klpd.Antenna>Antenna</klpd.Antenna>
<klpd.Collocationinf>Collocationinf</klpd.Collocationinf>
<klpd.FrequencyStd>FrequencyStd</klpd.FrequencyStd>
<klpd.HumiditySensor>HumiditySensor</klpd.HumiditySensor>
<klpd.PressureSensor>PressureSensor</klpd.PressureSensor>
<klpd.TemperatureSensor>TemperatureSensor</klpd.TemperatureSensor>
<klpd.WaterVaporSensor>WaterVaporSensor</klpd.WaterVaporSensor>
<klpd.OtherInstrumentation>OtherInstrumentation</klpd.OtherInstrumentat
ion>
<klpd.SurveyedLocalTies>SurveyedLocalTies</klpd.SurveyedLocalTies>
<klpd.Receiver>Receiver</klpd.Receiver>
<klpd.InstrumentType>Float</klpd.InstrumentType>
<klpd.InstrumentPrecission>10 mm</klpd.InstrumentPrecission>
```

```
</EQUIPMENT>

<MONUMENTINFO>
<klpd.BaseMonumentInfoLib>BaseMonumentInfoLib</klpd.BaseMonumentInfoLib
>
<klpd.FormInformation>FormInformation</klpd.FormInformation>
<klpd.MoreInformation>MoreInformation</klpd.MoreInformation>
<klpd.SiteIdentification>KLPD</klpd.SiteIdentification>
<klpd.SiteName>Klaipeda</klpd.SiteName>
<klpd.SiteCountry>Lithuania</klpd.SiteCountry>
<klpd.Contributor>Marine Research Center; VGTU GI</klpd.Contributor>
<klpd.Latitude>55.7165</klpd.Latitude>
<klpd.Longitude>21.1183</klpd.Longitude>
<klpd.Datum>Normaal Amsterdams peils (NAP)</klpd.Datum>
</MONUMENTINFO>

<LOCALINTERFACES>
<klpd.BaseLocalInterfaces>BaseLocalInterfaces</klpd.BaseLocalInterfaces
>
<klpd.RadioInterfaces>RadioInterfaces</klpd.RadioInterfaces>
<klpd.LocalEvents>LocalEvents</klpd.LocalEvents>
<klpd.StartDateAndTimeUTC>1949/01</klpd.StartDateAndTimeUTC>
<klpd.EndDateAndTimeUTC>2006/3</klpd.EndDateAndTimeUTC>
<klpd.RecordIntervalsInSeconds>600</klpd.RecordIntervalsInSeconds>
<klpd.SamplingIntervalsInMinutes>43200</klpd.SamplingIntervalsInMinutes
>
<klpd.QualityControlLevel>L1</klpd.QualityControlLevel>
</LOCALINTERFACES>

<NAMESPACES>
<klpd.BaseContactLib>BaseContactLib</klpd.BaseContactLib>
<klpd.PersonInChargeName>PersonInChargeName</klpd.PersonInChargeName>
</NAMESPACES>

<PARAMETERS>
<klpd.Parameter1>Date (yyy/mm)</klpd.Parameter1>
<klpd.Parameter2>QC Flag</klpd.Parameter2>
<klpd.Parameter3>Observed Sea Level (m)</klpd.Parameter3>
<klpd.Parameter4>QC Flag</klpd.Parameter4>
</PARAMETERS>

%End of Header
<polyline style="&sto18;"
      points="
0.2, -0.3020
0.4, -0.2960
0.6, -0.2970
0.8, -0.3040
1, -0.3170
<...>
```

Fig. 4. Excerpt from the SVG file

Still, hourly SVG files are formed with the help of the file written in PHP scripting language [9]. Data from the sensors is transmitted to the server, where MYSQL query is formed and put on the MYSQL database. When PHP file that forms SVG with XML Data Schemas is accessed, it takes last hour's means from database and forms corresponding SVG file, which can be saved to local hard drive [6, 9].

Currently there are compatibility issues when different browsers access formed SVG file. Plain SVG file is correctly interpreted by all three main browsers the files are tested on – *Internet Explorer*, *Opera*, *Firefox*. As for SVG file and implemented XML Data Schemas, all three of them interpretate differently: *Firefox* and *Opera* has built-in SVG viewers, while *Internet Explorer* uses *Adobe SVG Viewer*.

Structure of XML Data Schemas

XML Data Schemas are divided into separate logical groups of files. The main idea of such dividing was taken from [8–10]. On the server, XSD files are grouped on separate folders (Fig.5).

```
drwxr-xr-x 2 X X 4096 2007-03-21 13:42 common
drwxr-xr-x 2 X X 4096 2007-03-21 13:42 contact
drwxr-xr-x 5 X X 4096 2007-03-21 13:42 equipment
drwxr-xr-x 2 X X 4096 2007-01-02 16:56 localInterferences
drwxr-xr-x 3 X X 4096 2007-01-02 17:36 monumentinfo
```

Fig. 5. Folders of XSD files

When header is formed, responsible script applies to corresponding file in a folder, which then passes to the script information in XML Data Schema and so the part of header is formed. For example, excerpt from the site location file is given in Fig. 6.

```
<...>
<xsd:complexType name="siteLocationType">
  <xsd:sequence>
    <xsd:element name="city" type="xsd:string" />
    <xsd:element name="state" type="xsd:string" />
    <xsd:element name="country" type="xsd:string" />
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element                  name="xCoordinateInMeters"
type="xsd:string" />
          <xsd:element                  name="yCoordinateInMeters"
type="xsd:string" />
          <xsd:element                  name="zCoordinateInMeters"
type="xsd:string" />
          <xsd:element name="latitude-North" type="xsd:string" />
          <xsd:element name="longitude-East" type="xsd:string" />
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
    <xsd:element name="notes" type="xsd:string" />
  </xsd:sequence>
</xsd:complexType>
<...>
```

Fig. 6. Excerpt from the site location file

Here corresponding XML Data Schema values can be edited, deleted or added. While external form is under development, the only way is to replace values by hand, but it will be finished with the current final version of this SVG implementation in XML Data Schemas. These values are inserted into the main SVG format file. So all the header information, needed for the one, viewing SVG is all here, in human readable form and as it is XML based, it can easily be parsed with simple XML parser.

SVG file scripting

As for PHP file, forming SVG, there are simple *echo* statements that prints out to the SVG file all the collected header information, calculated SVG values and thus forms SVG file (Fig. 7).

```
$i=1;
$max=-1000;
$min=1000;
while($row=mysql_fetch_array($result)){
  echo "".$i.", ".$row['lygis']."\n";
  if($max<$row['lygis']){ $max=$row['lygis'] };
  if($min>$row['lygis']){ $min=$row['lygis'] };
  $i=$i++};
echo "\"/>\n";
$max++;
$min--;
$max=round($max);
$min=round($min);
```

Fig. 6. Excerpt from PHP script

Surely, if moving the file onto another server, PHP file must be edited by hand. Sensitive information as MYSQL database passwords etc. should not be passed other way. As for formatting SVG values we show you the cycle, which prints out last hours values. If talking about different viewBoxes (SVG attribute), variable *$i* might change accordingly. Everything else is self-explanatory. We choose maximum value of -1000 and minimum value of 1000 *($max, $min)*. So we wouldn't confuse extremely low maximum values and extremely high minimum (if any). The we print out the values, where *$row['lygis']* is the value from database. After that we check if the current *$row['lygis']* value fits as a maximum or minimum value and if this is the last time we go through this cycle, we close the brackets. So, SVG values are passed. As for *$max* and *$min* values, it we form the box for better viewing pleasure, where graph is shown.

Conclusions

1. Merging XML Data Schemas and SVG gives some advantages for sea level data exchange. The graphical representation of marine data is already working with flexible header formation system, so important for data exchange between various sites. All the same, this presented work is only for sea level measurements values, however, very easily it could be adapted for other marine data.
2. SVG file can be seen as graphic or as raw data, when viewed the source file. So it could be the base of the TGINEX format version 2, by what will improve the data exchange within the marine community.

References

[1] **Woodworth P. L., Player R.** The Permanent Service for Mean Sea Level: An update to the 21st century. Journal of Coastal Research Vol. 19 (2) (2003), p. 287–295.

[2] **Woodworth P. L.** The Permanent Service for Mean Sea Level. Journal of Geodesy Vol. 77 (10–11) (2004), p. 604.

[3] **Plag H.-P., Axe P., Knudsen P., Richter B., Verstraeten J.** European Sea Level Observing Systems (EOSS). Status and Future Developments. European Commission, Cost Action 40, EUR 19682 (2000), 26 p.

[4] **ESEAS.** Classification of ESEAS Operational Sites for Applications A-D (2004a), http://www.eseas.org/eseas-ri/deliverables/d1.3

[5] **ESEAS.** Standards for Quality Control of Tide Gauge Observations, Deliverable D1.2 (2004b), http://www.eseas.org/eseas-ri/deliverables/d1.2

[6] **Plag H.-P.** The ESEAS Data Portal: Principal Considerations, in: S. Holgate, T. Aarup (ed.). Workshop on New Technical Developments in Sea and Land Level Observing Systems, IOC Workshop Report No 193 (2004) p. 108–113.

[7] **Garcia M. J. et al.** European Sea Level Monitoring: Implementation of ESEAS Quality Control, in: Conference Dynamic Planet, Cairns, Australia, 22–26 August 2005. 8 p.

[8] **Parseliunas E., Viskontas P., Grigas M., Obuchovski R., Marozas L.** Measurement Techniques of a European Sea Level Service Site in KLAIPEDA. Solid State Phenomena Vol. 113 (2006), p. 483–488.

[9] **Parseliunas E., Urbsys A., Grigas M.** Computer Graphic Metafile for presentation of long-term sea level data on the internet, in: Selected papers of 6th International Conference “Environmental Engineering”, Vol. 2. Vilnius: Technika (2005) p. 969–974.

[10] Information on http://sopac.ucsd.edu/projects/xml/

[11] The development of a sea level metadata web service demonstrator project, in: Summary report, Intergovernmental Oceanographic Commission of UNESCO, IOC Project Office for IODE, Ostend, Belgium, 28-29 March 2006. 33 p.

[12] **Parseliunas E., Marozas L.** XML Data Schemas of the Sea Level Observations for Scalable Vector Graphics Format. Proceedings of the International Conference VIBROENGINEERING 2008: 128–130.

[13] **Sliogeris P.** An XML Based Marine Data Management Framework. Australian Oceanographic Data Centre. 12 p.

[14] **Ronai B., Sliogeris P., Plater M., Jankowska K.** Development and Use of Marine XML within the Australian Oceanographic Data Centre to Encapsulate Marine Data. Australian Oceanographic Data Centre. 27 p.