

UDK 551.461.2

XML DATA SCHEMAS AND SCALABLE VECTOR GRAPHICS FORMAT FOR THE SEA LEVEL OBSERVATIONS

Leonardas Marozas¹, Eimuntas Paršeliūnas¹, Saulius Urbanas²

¹*Institute of Geodesy, Vilnius Gediminas Technical University,
Saulėtekio al. 11, LT-10223 Vilnius, Lithuania, e-mail:gi@ap.vtu.lt*

²*National Land Service, Gedimino pr. 19, LT-01103 Vilnius, Lithuania, e-mail:sauliusu@zum.lt*

Abstract. XML (Extended Markup Language) Data Schemas as the format for information exchange in graphical SVG (Scalable Vector Graphics) format is presented. SVG format is a language for two-dimensional graphics and is based on XML. It is the advanced version of previous data exchange format TGX (Tide Gauge Independent Exchange Format), which contains the plain ASCII data. SVG graphics are formatted from sea level observations data at ESEAS station KLPD. Comparing with similar CGM (Computer Graphics Metafile) format, SVG has its advantages of being based on XML.

XML Data Schemas, in addition, add a header 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 an exchange between different sites from different countries so as it would be information standardisation. Visual appearance of the SVG file, the source, headers and formation of the file in script are described too.

Keywords: Data Schemas, XML, SVG, CGM, TGX, sea level observations, distribution of data.

1. Introduction

Data exchange for marine research sites is extremely important and valuable [1–8]. At present TGX format (only ASCII data) is used for sea level observations data exchange [9–10]. Since CSV (comma separated values) or SDF (standard data format) format save place and as raw textual data can be exchanged, interpreted and imported easily, therefore graphical representation of data remains complicated. As for machine, processing data is not a problem, for human being, working with the data; it is very hard to compare, maintain and estimate the data. After TGX format was implemented, next short step was the introduction of CGM format for marine data exchange [11]. However, because of some disadvantages, the complete level ahead of TGX data is though now currently under development, but already functioning SVG (Scalable Vector Graphics) file format with XML Data Schemas [12–15].

2. Structure of SVG file

SVG was introduced in 1999 as an open standard by the W3C. It is a language for describing 2D graphics in XML. It is used in many business areas like web graphics, animation, user interfaces, mobile etc. One of the main areas that uses SVG format for data representation is Geographic Information Systems, that have very specific requirements and that meets the SVG: rich graphic features, support for vector and raster content, ability to handle huge amounts of data etc. For example, graphical elements can be identified as their

native objects (lakes, seas, hills etc), allowing applications to interact with objects in a graphical manner. Three types of graphic objects can be represented via this format: vector graphics (lines, curves, shapes etc), images and text.

Text can be in any XML namespace, suitable for application, which enhances searchability and accessibility of the SVG graphics. The drawings can provide additional interactivity or dynamics. Animations of objects may be defined in two ways – via scripting (through languages as ECMAScript or SMIL) and by embedding SVG animation elements in content.

The main advantages of this format is its rather small size comparing with raster graphics image formats like *jpeg*. The images are scalable (not being limited to a fixed pixel size). It is greatly flexible because of vectoric representation of data (it can, as mentioned be combined with raster images). And the use of XML, which became very popular for structured information exchange. The main disadvantage would be the need of external SVG viewer for SVG in Web viewing in the most popular browser – Internet Explorer. Other browsers like Opera (version 9 and up) has full SVG 1.1 implementation, also browsers, based on Gecko layout engine version 1.8 (Firefox 1.5, Camino etc) also have full SVG 1.1 implementation.

There are many ways to use SVG in Web pages – as a stand-alone SVG web page, embedded by reference (image, object or applet tags in HTML), external link, using `<a>` element etc. Now about the structure of the SVG document.

SVG document fragment may consist of any number of elements contained within an `svg` element (elements – the same definition as in XML). The document fragment may range in its content from very simple or empty `svg` elements to really complex, deeply nested collection of container elements (elements that can have graphics elements and other container elements as child elements – *svg*, *g*, *defs* *symbol* etc.) and graphics element (that can cause graphics to be drawn onto the target canvas – *path*, *text*, *rect*, *circle* etc).

SVG document could be formed in the following way:

- XML definition.
- XML Data Schema. It includes the information about
 - Equipment: data format, PSMSL Code, antenna information, collocation information, frequency Standard, information about humidity, pressure, temperature, water vapour sensors, receiver information, instrument type, and precision;
 - Monument Info: form information, additional information, site identification, site name, site country, contributor, latitude, longitude, datum;
 - Local interfaces: base local interfaces, radio interfaces, local events, start date and time, last date and time (the time of data shown in generated document), record interval in seconds, sampling intervals in minutes and quality control level;
 - Namespaces: base contact information and person in charge information;
 - Parameters: these are 3 parameters, called correspondingly `parameter1`, `parameter2` etc. They are the date (yyy/mm), QC flag, the data measured and shown in corresponding document (eg observed sea level (m)).
- The data of measurements in human-readable format (eg 0,0120).

- Additional information (coordinate system, colours, the size of one step in graphic and other means, needed for correct formation of SVG document).

In this case, the document is fully self-described and humanly (and machinery) readable ie it might be parsed with some common tools that deals with standard text information etc.

As for now, it seems like one of the most progressive and easy-to-use-and-read format available for showing such complex and versatile information as marine measurements. This means that with the template, shown above, it is possible to describe almost every marine data possible.

The SVG file is composed from two main parts: XML Data Schemas and observation data. 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 of KLPD site is presented in Fig 1.

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 – thus XML Data Schema would be easily modified without actually having to reprogram anything.

The use of the XML Data Schemas gives some advantages for data exchange purposes [14]. First, its great strength is that XML Data Schemas are written in XML, so it becomes easier to parse data with a simple XML parser, then comes the security aspect of XML Data Schemas, extensibility, readability etc. Second, XML gives an opportunity to use SVG as vector graphics format - very powerful technology to visualise the technological data without losing 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 2. Excerpt from the simple source code of SVG file, representing the results of sea level observations, is given in Fig 3.

```
<...>
<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 1. Excerpt from the XML Data Schemas header

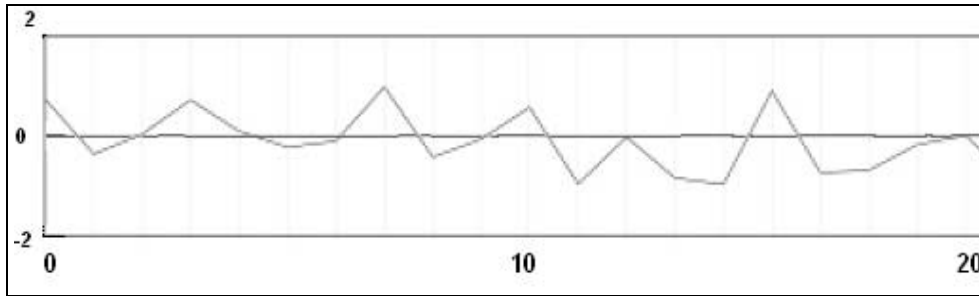


Fig 2. Sea level hourly means data graphical representation in SVG file. First 20 minutes

1,	0.74
2,	-0.36
3,	0.04
4,	0.73
5,	0.1
6,	-0.22
7,	-0.1
8,	0.99
9,	-0.42
10,	-0.06
11,	0.58
12,	-0.96
13,	-0.02
14,	-0.84
15,	-0.97
16,	0.91
17,	-0.73
18,	-0.69
19,	-0.17
20,	0.01

Fig 3. First 20 results of sea level observations

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 a corresponding SVG file, which can be saved to local hard drive [7, 9].

Currently there exist 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* have built-in SVG viewers, while *Internet Explorer* uses *Adobe SVG Viewer*.

3. Structure of XML Data Schemas

The main advantage of XML Data Schemas is their transformability which means that it can be transformed between XML based documents, complying to different schemas, ie we convert geographical data (GML) to 2D

graphics (SVG) for better “readability”, thus that plain data could be shown graphically (eg a map).

An XML Data Schema is a wellformed XML document and must be a valid instance of XML Schema DTD (Document Type Definition. It defines legal building blocks by an XML document.). The purpose of this scheme is to define legal building blocks like DTD. Thus it:

1. defines elements that can appear in a document,
2. defines attributes that can appear in a document,
3. defines which elements are child elements,
4. defines the order of child elements,
5. defines the number of child element,
6. defines whether an element is empty or can include text,
7. defines data types for elements and attributes,
8. defines default and fixed values for elements and attributes.

In fact, XML Data Scheme is the successor of DTD and there are hopes that it will really soon replace DTD in most Web Applications, where DTD is used now.

The main requirements for correctly formed XML Data Scheme are:

1. must begin with the XML declaration,
2. must have one unique root element,
3. all start tags must match end-tags,
4. XML tags are case sensitive,
5. all elements must be closed,
6. all elements must be properly nested,
7. all attribute values must be quoted,
8. XML entities must be used for special characters.

The very beginning of XML Data Schema must define all the namespaces, used in schema. The namespace for Schema language itself is defined by URN (Uniform Resource Name, a subset of Uniform Resource Identifiers (URIs)), using namespace declaration.

XML Data Schemas are divided into separate logical groups of files. The main idea of such division was taken from [10]. On the server, XSD files are grouped in separate folders (Fig 4). 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 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 4. Folders of XSD files

```

<...>
<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 5. Excerpt from the site location file

Here corresponding XML Data Schema values can be edited, deleted or added. While an 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 a simple XML parser.

4. 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 6).

```

$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 be not passed in 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*). Thus we would not confuse extremely low maximum values and extremely high minimum (if any). Then 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. Thus, SVG values are passed. As for *\$max* and *\$min* values, they form the box for better viewing pleasure, where graph is shown.

5. 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. Thus could be the base of the TGX format version 2, which will improve the data exchange within the marine community.

Literature

1. WOODWORTH, P. L.; PLAYER, R. The permanent service for mean sea level: An update to the 21st century. *Journal of Coastal Research*, 2003, 19 (2), p. 287–295.
 2. WOODWORTH, P. L. The permanent service for mean sea level. *Journal of Geodesy*, 2004, 77 (10–11), 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 (2004a). Classification of ESEAS operational sites for applications A–D. Available from Internet: <http://www.e seas.org/e seas-ri/deliverables/d1.3>
 5. ESEAS (2004b). Standards for quality control of tide gauge observations, Deliverable D1.2. Available from Internet: <http://www.e seas.org/e seas-ri/deliverables/d1.2>
 6. ESEAS Governing Board. European sea level service. terms of reference. 2001 (draft). Available from Internet: <http://www.e seas.org/websitemap.php>
 7. PLAG, H.-P. The ESEAS data portal: principal considerations. In Holgate, S.; Aarup, T. (ed.). *Workshop on New Technical Developments in Sea and Land Level Observing Systems*, IOC Workshop Report No 193, 2004, p. 108–113.
 8. 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.
 9. PARSELIUNAS, E.; VISKONTAS, P.; GRIGAS, M.; OBUCHOVSKI, R.; MAROZAS, L. Measurement techniques of a European sea level service site in KLAIPĖDA. *Solid State Phenomena*, 2006, Vol 113, p. 483–488.
 10. PARSELIUNAS, E.; VISKONTAS, P.; GRIGAS, M.; OBUCHOVSKI, R.; KOLOSOVSKIS, R.; BARONAS, T. Technical equipment and software of European sea level service site KLAIPĖDA. In *Selected papers of 6th International Conference “Environmental Engineering”*, Vol 2. Vilnius: Technika, 2005, p. 964–968. ISBN 9986–05–851–1.
 11. PARŠELIŪNAS, E.; URBŠYS, 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. ISBN 9986–05–851–1.
 12. Information on <http://sopac.ucsd.edu/projects/xml/>.
 13. *The development of a sea level metadata web service demonstrator project*. Summary report, intergovernmental oceanographic commission of UNESCO, IOC Project Office for IODE, Ostend, Belgium, 28–29 March 2006. 33 p.
 14. SLIOGERIS, P. *An XML based marine data management framework*. Australian Oceanographic Data Centre. 12 p.
 15. 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.
-
- Eimuntas Paršeliūnas**. Assoc Prof, Doctor. Vilnius Gediminas Technical University. Dept of Geodesy and Cadastre, Ph +370 5 274 4703, Fax +370 5 274 4705.
Doctor (1992). Author of two teaching books and more than 45 scientific papers. Participated in many intern conferences.
Research interests: graphs theory in geodesy, adjustment of geodetic networks, geoinformation systems.
-
- Leonardas Marozas**. Bachelor degree student. Vilnius Gediminas Technical University. Dept of Geodesy and Cadastre, Ph +370 5 274 4703, Fax +370 5 274 4705.
Author of two scientific papers.
Research interests: computer operation systems, computers networks, geoinformation systems.
-
- Saulius Urbanas**. Doctor. National Land Service under the Ministry of Agriculture. Director of Dept of Geodesy and Cadastres, Ph +370 5 2398440, Fax +370 5 2391331.
Doctor (2001). Author of 8 scientific papers. Participated in many intern conferences.
Research interests: object orientated geoinformation systems, generalisation in cartography.