XML applications in SLR

Kalvis Salminsh
Institute of Astronomy
University of Latvia
Boulevard Rainis 19
LV 1586 Riga, Latvia

kalvis@lanet.lv

Abstract
The SLR data analysis and management software, developed by the author for the new SLR system at GFZ-Potsdam, uses relational database to store almost all data used at the station, including range measurements, predictions, station coordinates, generated normal points etc. Internally in the database some data, most notably the ranging measurements, are stored in the XML (eXtendable Markup Language) based formats. The XML and associated techniques such as a XSLT (Extensible Stylesheet Language Transformations) allowed to design a flexible data format for the ranging data which supports future extensions, automatic verification and conversations to other formats including HTML, plain text etc. for publishing or for the use in the other software. Some other possible applications of the XML based data formats for the SLR data are mentioned as well.

Introduction.
The new data analysis and management software for the SLR station at the GFZ-Potsdam is designed as a client-server application to support a multi-user environment, remote data access and facilitate future station automation. According to this concept all station related data are stored in the relational database to avoid unnecessary data duplication and to offer centralized access from other computers. One of the problems, faced during the development was how to organize and store nontrivial data structures such as ranging measurements, orbital elements, TIRV's (Tuned Inter-range Vector) to name a few, in the relational database and to avoid or minimize the changes in the database structure and software when changes occur in the station's hardware, additional parameters are registered during tracking, incoming data e.g. predictions are modified, new requirements for the data publishing are issued, etc. Most of these data entities outside database usually can be represented as files (ranging data for particular pass, set of orbital elements, TIRV for one day). It was decided to store them inside database as a whole in the table fields. In most cases these data entities have a quite sophisticated internal structure, which renders commonly used simple text file format (rows of delimited numbers) in our opinion ineffective and hard to maintain, especially if additional information should be added later or in case of some other modifications or for the data exchange. As a solution, the usage of the XML based formats was selected. XML are now widely used in the scientific community for the data representation and exchange: MathML for describing and exchanging mathematical expressions, XDF – common scientific data format developed at NASA (see http://xml.gsfc.nasa.gov), Chemical Markup Language (CML™), to name a few. See http://www.xml.com for the brief introduction on XML and [1], [2] for more detailed information about the XML and associated technologies. XML support is now freely available virtually in all major operating systems and programming languages. As a result the new data formats for the ranging raw and filtered data were developed for the use in the software system at the station. Other main uses of XML based

techniques are universal prediction format for the analysis program (currently under development) and for the reports.

XML format for ranging data

Let us look on the raw data format as an example of the XML usage. The raw data file format was designed as a self-contained unit containing all recorded information about the tracked satellite, including station hardware data, and may be extended with new data and are stored in the database as an entity. See Appendix A for the example of the ranging data in the old format and the same data represented as an XML in the Appendix B. In the XML version the information is organized in the tree like structure using named tags. Each tag may have one or more attributes with numerical or textual values. The raw data is divided into the two main parts, marked with the "Header" and "Data" tags. The header part contains information about the satellite, station, hardware used, calibration and meteorological data. Each logical section is marked with the corresponding pair of tags or single tag. The calibration and meteorological data may be represented as a pre- and post-pass values or as a data tables which later can be interpolated. The data section contains a set of "DataPoints" elements. Each element represents one measurement and the recorded values are stored in the attributes. While from the first glance the only difference between two versions is more descriptive free format look of the XML version, the principal differences are that the information is presented in a structured way and in the document handling. Using XML manipulation tools or programming interfaces it is easy to transform document to other formats or modify it and it can be done without requiring changes in other programs that use these data. For instance, for each range measurement the attribute for the signal amplitude can be added later, but existing analysis program doesn't know about it but will continue to work without any changes. From the other hand, some other programs may take an advantage of using this value when working with this data. It's also possible to write a formal description (XML Schema) of the document structure and using this definition to verify the document structure and content with the help of the external tools or inside the script or program. This verification feature also allows performing input data validation in a much cleaner way, without writing a multitude of the conditional checks. Another useful feature of the new format is a possibility to add a comment to any place between the tags. The program, producing raw data for analysis, can annotate data if necessary and the filtering program also can additional comments, if necessary. The filtered data uses the same structure as a raw data; only calculated azimuth and elevation data is added to each range measurement. Using XSLT transformations the ranging data can be easily converted to other formats. See Appendix C for an example. It should be noted that the script is operating system independent and should run on any system with the XML support.

Other applications.

XML is often regarded as a web-only related technology, but one of its main strength is the data exchange between different computers and software systems. One possible application for the SLR community, where XML seems to fit well, is a prediction distribution and station outgoing data: normal points and full rate data. Changing existing formats and data distribution ways without a good reason obviously is not feasible, but in the case of new developments XML based technologies deserves an attention. Discussing all related issues will be well out of the scope of this article, but in our opinion the main advantages of the XML are:

- The flexibility of the XML allows overcome obvious limitations of the currently used data formats and better supports transition in cases when changes will be necessary.

- Schema definition language can be used to describe and publish formats such as full rate data and to validate data at the stations and data centers.
- Adaptation of XML will enable the use of already existing technologies such as a SOAP (Simple Object Access Protocol), defined by the World Wide Web Consortium, providing an universal data format readily adaptable to network communications, regardless of the operating system or programming language, can improve SLR data publishing and it access to the other users.

Conclusions

- The transition from fixed text format files to the XML based format for the ranging data eases data management and offers much higher flexibility
- XML based data facilitate data publishing and exchange between different programs
- Working with XML data in most cases is easier than in comparison with the plain text based data, especially from the programming point of view.
- The increased volume of the XML data compared to fixed format is not important for SLR
- Compatibility issues with older software can be resolved with scripts and XSLT transformations

Acknowledgments

The work was supported by GFZ –Potsdam under contract 102504-C

References

[1] S. Holzner "Inside XML", New Riders Publishing, 2000

[2] S. Holzner "Inside XSLT", New Riders Publishing, 2001

Appendix A.
Example of the raw data file in plain text format.

52322
0105604
754.4854
1.52
77.8
-3.24966669082642,
0.168024175859679
175
0
0
0
53.7107672691345,
6.56976252794266E-02
78365801
0
0
0
0
0
0
0
0
77656.1002512
10380487.115
77656.2002517
10377389.844
77980.0002626

…

11176716.694
77983.2002602
11284274.466

Appendix B
Example of the raw data in XML format.


```
<SLRRawData Version="1" Filtered="No" >
<!--Created by SLR Utilities module on 3/7/02 9:57:50-->
<Header >
<Epoch MJD="52322"/>
<Satellite Cospar="0105604" />
<Station SOD="78365801" SCH="8" SCI="3" CalibMethod="0" Epoch="4"/>
<Hardware>
<Laser Wavelength="5320" Const="0.9488" N1="-4" N2="4"/>
<PMT Voltage="1000"/>
</Hardware>
<Calibration>
<PrePass Epoch="77656.1002512" Value="53.7107672691345 RMS="6.56976252794266E-
02" />
<PostPass Epoch="77983.2002602" Value="-3.24966669082642"
RMS="0.168024175859679"/>
<Data Num="0"/>
</Calibration>
<Meteo>
<PrePass Epoch="77656.1002512" T="1.52" Hum="77.8" Pre="754.4854"/>
<PostPass Epoch="77983.2002602" T="1.52" Hum="77.8" Pre="754.4854"/>
<Data Num="0"/>
</Meteo>
</Header>
<Data NumObs="175" >
  <DataPoint Id="0" Epoch="77656.1002512" Range="10380487.115"/>
  <DataPoint Id="1" Epoch="77656.2002517" Range="10377389.844"/>

          ...

  <DataPoint Id="173" Epoch="77980.0002626" Range="11176716.694"/>
  <DataPoint Id="174" Epoch="77983.2002602" Range="11284274.466"/>
</Data>
</SLRRawData>
```

Appendix C.

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
xmlns:fo="http://www.w3.org/1999/XSL/Format">

<xsl:output method="text" encoding="utf-8"  />
<xsl:strip-space elements="*"/>

<xsl:template match="/">
<xsl:apply-templates />
</xsl:template>

<xsl:template match="/SLRRawData/Data/DataPoint" >
<xsl:value-of select="@Epoch" />
<xsl:text> </xsl:text>
<xsl:value-of select="@Range" />
<xsl:value-of select="'&#10;'" />
</xsl:template>

</xsl:stylesheet>
```

Figure 1: XSLT script to extract all epoch and range values from the raw data file to produce simple table

```
77656.1002512 10380487.115
77656.2002517 10377389.844
77658.7002533 10300346.004
77659.3002533 10281951.202
77659.4002549 10278906.831
77660.400255 10248353.035
…
```

Figure 2: XSLT script output fragment – epoch and range values