University of Massachusetts Lowell
Center for Atmospheric Research
600 Suffolk Street
Lowell, MA 01854

Phone : 978 934 4900
Fax : 978 459 7915

http://ulcar.uml.edu

UMass Lowell
Center for Atmospheric Research

Memo

Date: October 19, 2005

To: INAG discussion of SAOXML format

From: Bodo Reinisch, Ivan Galkin, Grigori Khmyrov, Alexander Kozlov

Cc:

RE: WDC-A Comments on Ionosonde Data Exchange Format – Draft

The SAOXML-5 format development team at the University of Massachusetts gratefully acknowledges the WDC-A draft comments on the format by T. Bullett, R. Redmon, and R. Conkright in Boulder, CO, published in the INAG Bulletin on 10 October 2005. Their document presents an in-depth analysis of the proposed SAOXML-5 format and suggestions for its modification. We have also received several other contributions containing critique of and suggestions for improvement of the original SAOXML proposal. We are pleased to see this degree of interest from the ionospheric community to the data format that will become the connecting link for a multitude of users and producers of ionosonde data.

The proposed WDC-A modifications have been reviewed, and in this memo we would like to discuss them through the prism of one important assumption: our fellow ionosonde data users will eventually write the SAOXML format readers and writers to use in the software they develop. If the format is clear and easy to work with, this will be easy to do. Standard XML libraries and sample SAOXML code in the appropriate language can be used for inspiration. It is ultimately important, therefore, that the format presentation is clear, internal architecture is simple but flexible, and support of the basic operations (reading, adding, removing, changing, skipping data elements) is robust – **without intricate software engineering**.

1. Presentation and Naming

The Boulder proposal makes a significant contribution to clarity and readability of SAOXML format specifications. We welcome many of the changes to the names of the data elements and the name capitalization convention. We have updated the UML proposal, accommodating many of the suggestions made by the Boulder group as well as the COST296 group (R. Stamper, U.K) and L.A

McKinnell for the South African Ionosonde network). The proposed revised version of SAOXML-5 is attached.

We do have reservations about one suggestion to rename "standard URSI *ionospheric* characteristics" to "standard URSI *ionogram* characteristics". We prefer to keep the original URSI terminology to avoid confusion. After all, the list is a mixture of the ionospheric (e.g., TEC, peak layer heights, scale heights) and ionogram characteristics (e.g., h'F2), and there are several popular ionogram-scaled values that have found their permanent place in ionospheric models. If we are to enforce proper naming, we shall then separate the URSI table in two parts, "ionogram" and "ionospheric", which seems unnecessary. In accordance with the pioneering formatting effort of URSI's Ionospheric Informatics Working Group, the new **SAOXLM format does not specify the *ionogram* format**.

2. XSIL Library

The XSIL syntax and nomenclature is used extensively in the Boulder proposal for storage of arrays. For comparison, Listing 1 and 2 show how the same trace information is presented in the two arrangements.

Listing 1: Digisonde Trace Example (storage by column)

```
<TraceList Num="1">
<Trace TraceName="1F2" Polarization="O" Num="9">
<Frequency Units =" MHz">3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 </Frequency>
<Range Units ="km">232.5 233.75 235.0 225.0 230.0 230.0 230.0 260.0 485.0 </Range>
<Amplitude Units ="dB">106 0 106 106 102 102 102 102 96 </Amplitude>
<Doppler NoValue ="99" Units ="Hz"> -.391 99 .391 -.391 .391 -.391 .391 .391 -.391 </Doppler>
</Trace>
</TraceList>
```

Suggested Boulder format:
Listing 2: Digisonde Trace Example (storage by XSIL table)

```
<TraceList Num="1">
<Trace TraceName ="1F2" Polarization = "O" NumColumns ="4" NumRows ="9">
<Column Name="Frequency" Type="Float" Unit="MHz" SigFigs ="5" Description="Nominal Frequency" />
<Column Name="Range" Type="Float" Unit="km" SigFigs ="5" Description="Range" />
<Column Name="Doppler" Type="Float" Unit="Hz" SigFigs ="3" NoValue="99.0" Description ="Doppler Shift" />
<Column Name="Amplitude" Type="Float" Unit="dB" SigFigs ="3" NoValue="0.000" Description ="RelativeAmplitude" />
<Stream Delimiter =",">
3.3000, 232.50, -0.391, 106.
3.4000, 233.75, 99.0, 0.000
3.5000, 235.00,  0.391, 106.
3.6000, 225.00, -0.391, 102.
3.7000, 230.00,  0.391, 102.
3.8000, 230.00, -0.391, 102.
3.9000, 230.00,  0.391, 102.
4.0000, 260.00,  0.391, 102.
4.1000, 485.00, -0.391, 96.0
</Stream >
</Trace >
</TraceList >
```

Presentation by the XSIL table looks better organized, be it at the cost of a significantly higher the data volume overhead. However, we note that the XSIL storage style **requires additional coding effort in the reader software** if it is to be upward compatible with future releases of the format containing new types of information.

For illustration purpose, lets consider an outdated SAOXML reader software that is unaware of the new possibility to store the Doppler frequency shifts of the trace echoes. When such reader encounters unknown data element <Doppler> in Listing 1, the XML parser can simply skip the

whole element. In Listing 2, when XML parser encounters the <Column> element with unknown Name="Doppler", it can also skip it safely, but then it will also need to separate, for each of the lines inside the table stream, tokens #3 corresponding to the unknown column position #3 and discard each of them. This operation becomes more intricate when there is more than one unknown column, and implementation of column mapping might be required to arrange proper skipping. As the XSIL project is no longer active, this coding has to be implemented directly in each user's code, in the user application language.

Implementation difficulty: light.

3. Significant Figures

It has been customary to use number of digits past the decimal point in all previous releases of SAO format, following the FORTRAN convention for formatting the floating point numbers. Concept of significant figures introduced in the Boulder proposal is a better choice, in light of recent demand for specification of ionosonde data uncertainties, but it must be clear that proper support of significant figures formatting has to be carefully coded in the SAOXML writer applications.

Implementation difficulty: light.

4. XML Attributes versus XML Elements

The Boulder proposal moves a few attributes of the <SAORecord> element (i.e., timestamp, station constants, scaler type, etc.) inside of the <SystemInfo> element and formats them as separate elements. Moving these attributes promises a higher degree of flexibility is achieved in terms of

- possibility to have more than one instance of the item (e.g., <StartTime> can be present more than once, in various time zones and formats) and
- better possibility of elaborating and expanding the item's structure and content.

We use attributes instead of elements for simplicity and efficiency of data access. The following analogy might help appreciate our original design: attributes are written on letter envelopes, elements are inside the envelopes (they require additional unpacking and syntax checking operations.) Scanning one-day file of the SAOXML records is faster and easier if we have the key descriptive information (time, station, type, manual/autoscaled) written on the envelope label. The postmaster shall not be required to open the envelope, find a particular set of pages, find among them a particular page labeled <Address> and then search for a particularly labeled address among other addresses on the same page.

Furthermore, we believe that it is a dangerous practice to allow flexibility in specification of the key pieces of information. The letter with unusually written address will not get to the destination. There must be a timestamp with UTC in the agreed standard, one and only format. A variety of

timestamps written in non-standard formats can be added to <SystemInfo> section if this were desired.

Finally, since ionosondes usually do not measure their latitude and longitude, and the gyrofrequency, and start time is also not one of the ionosonde measurements, we don't see a semantic issue with storing key information as attributes. Current list of these attributes includes

- FormatVersion
- StartTimeUTC
- URSICode
- StationName
- GeoLatitude
- GeoLongitude
- Source
- ScalerType

Implementation difficulty of parsing elements instead of attributes: light.

5. Timestamp

Use of the ISO8601 standard to form the timestamp is a good suggestion. Considering substantial heritage of providing day of year in ionospheric data, we suggest to use an extended form of the ISO8061 standard that includes DOY with a leading dash:

```
"2000-02-01 –032 13:45:05.000"
```

6. Hierarchical structures and associated management of pointers

Use of **pointers** instead of **values** is one particular technique that has caused objections by the ionosonde data users. Much caution has to be exercised with the multi-layered (hierarchical) constructions featuring links from layer to layer (i.e., data elements *pointing* to another element where the value shall be found). Such data organization requires a tender care to be coded in software correctly: hierarchical data types are rudimentary relational databases, and their proper management shall include protection of data integrity against operations of addition, removal, change, and skipping of elements constituting the data.

The SAO-4 Doppler Table is one example. Instead of spelling out a Doppler frequency value of, say, "-0.488" Hertz, all releases of SAO format prior to SAOXML 5 used to enumerate Doppler frequencies, put them in a table, and store pointers to the table instead of the value itself (see Table 1).

Table 1. SAO-4 Doppler Table

DOPPLER = "1"

ID	Doppler Frequency, Hz
0	-0.684
1	-0.488
2	-0.293
3	-0.098
4	0.098
5	0.293
6	0.488
7	0.684
9	NO VALUE

The Doppler Frequency Table is **no longer used in SAOXML.** It was a space-saving concept whose incurred complexity did not suit well with producers and consumers. We shall point that the Doppler table was a relatively simple object: it did not have to support operations of addition, removal, change, or skipping of the items.

There are two SAOXML revision teams that advocate other types of a multi-layered structured design involving pointers. Both suggested constructions are far more complex than the SAO-4 Doppler Table. In particular, the WDC-A team suggests the following hierarchy of dependencies for the trace information:

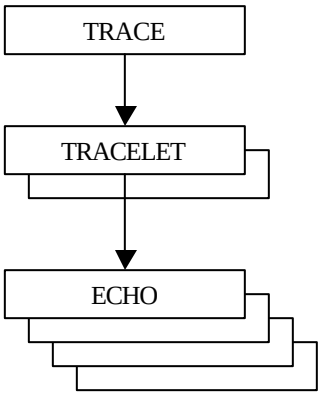

Figure 1. Multi-layered hierarchy for storage of trace data, Boulder proposal.

The trace is a set of pointers to tracelets, each tracelet being a set of pointers to echoes. This data organization is different from the mainstream design (Listing 1 and 2) found in other software projects, including the ARTIST, SAO Explorer, and DIDBase at UMLCAR. It would be only fair to give it careful examination in order to understand its advantages and associated overhead.

The multi-layered (hierarchical) organization is a strict construction that recognizes the fact that traces may consist of more than one element, "tracelet", whereas the tracelet is in fact a group of echoes, each echo having a set of attributes such as frequency, range, amplitude, Doppler frequency, etc. By building a pyramid of layers with the inter-layer links, certain structurization elegance is accomplished without adding or subtracting useful information. It is our opinion, though, that maintaining the referential integrity through everyday operations of reading, adding, removing, changing and skipping trace information would become considerably more difficult with introduction of this concept, and this drawback may outweigh the advantages.

To illustrate our concerns, let us first consider a simpler data item, a scaler name. Applying the multi-layered approach, we arrive at the architecture shown in Figure 2:

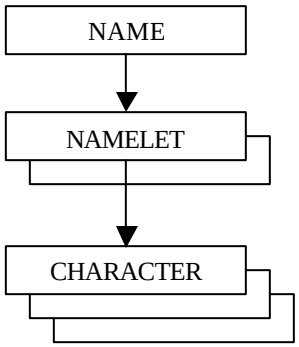

Figure 2. Multi-layered hierarchy for storage of names.

Thus, "John Johnson" name is a pair of pointers, one pointing to the namelet "John" and the other one to "Johnson", each namelet being a set of pointers to individual letters of the alphabet. Each letter has its unique ID, a letter value ranging from 1 to 26, and a number of other useful values such as the upper/lower case, font face, size, etc. There is considerable overhead in storing this construction, but it is not excessive.

One visible disadvantage, though, is that a human cannot read the name anymore; a computer is required to follow all links and assemble the name. Similarly, if we need to look at, say, virtual heights of the F2 region trace, O component, eye-balling the XML file for reasonable h'(f) traces would not be possible. We need to walk through the hierarchy: (1) locate reference to F2o trace in Traces, then (2) follow the pointer to the appropriate tracelet, (3) collect echo indices, follow each pointer to the pool of Echoes, (4) pick up the virtual height value from each found Echo, fill the array.

Whereas the English alphabet in Figure 2 will not change in the process of editing names, it is reasonable to assume that manual editing of traces can modify individual echoes. This warrants testing of the hierarchical pyramid against the ARCS operations (Addition, Removal, Changing, and Skipping). This analysis clearly shows involved additional overhead of enforcing referential integrity. Every deletion (of an echo, a tracelet, or a trace) needs to be carefully analyzed whether it

had disturbed the pyramid where all layers are interconnected. We will have to answer questions like, "are there any pointers referring to a missing data element on the underlying level" and "are there data elements that nobody points to from the upper level".

We are not saying that robust ARCS operations on the pyramid are impossible. They are possible, with certain effort. **This effort will have to be made by the user** as there are no existing ready XML solutions that we are aware of[1]. It is a question of justifying such proprietary development effort in the suite of user applications that will have to read or create SAOXML records.

The Boulder team lists the following advantages of the suggested hierarchical organization:

A1. "[It] better represents the physical realities of ionospheric sounding, data analysis and interpretation both past, present and future."

A2. "It allows for much better and formal error analysis and future data re-evaluation without going back to the raw ionogram records."

A3. [paraphrasing] It allows for the fact that some ionogram traces have well defined nomenclature, and other traces have evolving definitions and their value is a matter of ongoing research.

A4. [paraphrasing] It is quite flexible to support, in a unified fashion, other ionosonde data such as true height profiles and digisonde skymaps.

Our naïve example with hierarchical representation of a person's name helps to rule out advantages A1 and A4. It is true that hierarchical representation is universal and it reflects the fact that real life objects consist of smaller parts, but highlighting this fact does not add useful information in many cases, including storage of the ionogram traces. Storing of the trace data conventionally does impose requirement of a well-defined nomenclature, so we don't see a use for A3. We need further clarification of the advantage A2.

Implementation difficulty of hierarchical data types: heavy, proprietary.

7. Short-form XML names

Supporting short-form versions for all full-length names of elements and attributes looks like a requirement for a particular user, perhaps the US Government. By making this requirement a part of the SAOXML standard, we essentially extend it to all ionospheric community that now have to double the size of available keywords and clutter the code with additional syntax checks. It also looks like this requirement regulates the choice of full-length names in order for the short-form names to be unique. Both drawbacks will go away if we reconsider the need of the short-form naming to be standardized.

[1] Development of integrity constraints for XML at the node level is underway, and there are chances that solutions will be available in the future.

8. Summary

We have reviewed the modifications to the SAOXML format suggested by the Boulder team. Many of them were good and we incorporated them in the revised SAOXML-5 format that we have attached. Other suggestions, mainly those aiming of putting raw ionograms and skymap data into SAOXML, were analyzed in this report, but we have not incorporated them in the attachment. They would require additional development of proprietary software in order to use the format for reading and writing ionosonde data. We believe that ionosonde data users will implement their software interface for direct manipulation of data in SAOXML format using example code and standard libraries- in the same way they do it today with SAO-4. It is therefore important that we lighten the design that the Boulder group suggests, as it carries a number of heavy weight additions to version 5 that will indeed make most people turn away and resort to converters. This may be a direct way to torpedoing the project whose original purpose was to **make SAOXML human-readable in an Internet browser so that it is truly user-friendly.** Software development is needed, but it needs to be done only once, if it is done in such a way that further releases do not disturb operation of the upward compatible software.