IGS Site Log XML Teleconference 2016-11-14

Present: Fran B, Erwin W., Roger F, Nick B, Lazar B, Markus B.

Topics:

- Can we consider schema modifications based on EPN/ROB experience and/or workflow to handle the conflicting cases
- What comments do participants have on the content at geodesyml.org
- Code developments at Geosciences Australia that will be made available open source - any guesses on release date(s)
- Revisit our earlier timeline for making progress
- Others doing implementation?

Discussion:

We took the topics out of order from the proposed topics list.

1. Code developments at Geosciences Australia that will be made available open source - any guesses on release date(s)

Nick and Lazar gave a brief update of the work at GA.
The GeodesyML site and the Github repository are the two primary sites from which GeodesyML resources will be made available.

GA (Lazar) is working on Python code for Ascii site log -> GeodesyML Site Log XML encoding. This will be available for all in future.

2. Can we consider schema modifications based on EPN/ROB experience and/or workflow to handle the conflicting cases.

Lazar walked us through the comments that Carine submitted in response to the request for updates to the IGS-DCWG list. EPN/ROB has invested time in translating their existing site logs to the previously discussed schema. The comments from Carine are below, with the proposed change to the site log xml in blue.

We have been trying to map the site logs of the EPN and EPN densification (about 1300 site log files) in the GeodesyML format and our main concerns are related to (hoping we got it all right):

- fields that are mandatory in GeodesyML (mostly expecting a double) and are not in the site log.

In version 0.4 of GeodesyML, we intend to make all fields that are optional in ASCII site logs also optional in GeodesyML by marking them as nillable. For example, the following definition of elevationCutoffSetting:

<element name="elevationCutoffSetting" type="double" nillable=”true”/>

would allow GeodesyML documents to contain:

<GnssReceiver>
…
<elevationCutoffSetting xsi:nil=”true’’/>
…
</GnssReceiver>

- fields in the site log that have values such as "<5" while GeodesyML only accepts a double

GeodesyML enforces strict data type checking. At GA, we are stripping less-than signs, as well as units. For example, “<1m” becomes 1.0. We are keeping track of where such conversions have taken place to help diagnose any errors the conversions may introduce.

- TIME/DATE format incomptability:
GeodesyML accepts CCYY-MM-DDThh:mm:ssZ or CCYY-MM-DD
Site log accepts CCYY-MM-DDThh:mmZ or CCYY-MM-DD

The XML date and time data type requires seconds. See: http://www.datypic.com/sc/xsd11/t-xsd_dateTime.html. At GA, we are appending 00 seconds to our times.

- Site log section 3:
Our site logs have two fields (deg C) +/- (deg C) for the temperature stabilisation info from site log, while GeodesyML has only one. But we discussed this already in a previous mail.

- Site log section 6:
GeodesyML requests a double as input to "Input Frequency" even if an internal frequency is used.

We will make input frequency optional in the schema and implement a schematron rule to ensure that the frequency value is required only when an external clock is present.

- Section 8.5.1:
GeodesyML asks for a date input, which is not standard foreseen in site log

We will ensure that these dates are optional.

- Site log Section 11:
We choose to put site log Section 11 in 'siteContact' and additionally, put the first contact also in put in mandatory 'siteMetadataCustodian'.

We are putting Section 11 into siteContact, and Section 12 into siteMetadataCustodian. If Section 12 is blank, then we are also putting Section 11 into siteMetadataCustodian.

- Site log Section 12:
GeodesyML allows for only one contact person in 'siteOwner', while Site log allows multiple.

Site owner is one in GeodesyML, but site contacts can be many. Could you put your secondary contacts from ASCII site logs into siteContacts in GeodesyML?

- Site log Sections 11 and 12:
Fields: Prefered Abbreviation and Additional Information are missing in GeodesyML. So information is lost when converting to GeodesyML
and
The address in GeodesyML is split in deliveryPoint, city, postcode, country while in the site log, these fields are not always separated. Impossible to complete GeodesyML automatically. Need to split also fields in site log to accommodate GeodesyML.

1) GeodesyML adopts a "responsible party" data type defined in GMD schema, which does not include a preferred abbreviation attribute and additional information. If necessary, we could potentially look into extending the responsible party data type to include extra fields.
2) Yes, it's impossible to automatically split out addresses from site logs into the individual address fields in GeodesyML. At GA, we are initially inserting the entire address into the single deliveryPoint field.

In addition, we would like to see an extension of GeodesyML to allow providing the url to the file with individual antenna calibrations, if available.

This is already possible using the RemarksGroup element group. Element GnssAntenna inherits the following thee elements in RemarksGroup (commonTypes.xsd): remarks, associatedDocument, and extension. For example,

```
    <geo:gnssAntenna
gml:id="GNSS_ANT_1">
      <geo:associatedDocument>
        <geo:Document gml:id="
GNSS_ANT_1_CALIBRATIONS">

<geo:type>TEXT</geo:type>
          <geo:body>
            <geo:fileReference
xlink:href="https://..."/>
          </geo:body>
        </geo:Document>

</geo:associatedDocument>

<geo:manufacturerSeri
alNumber>CR20020709
</geo:manufacturerSeri
alNumber>
            …

</geo:gnssAntenna>
```

The telecon participants agreed to the idea that the changes above will be incorporated into a version 0.4 of GeodesyML. There will likely be a need for additional changes that will be recognized as the implementation by participants moves forward.

3. What comments do participants have on the content at geodesyml.org

Erwin reported on his initial look at the site and his attempt at bringing up Geoserver. He ran into difficulties and didn't have time to further work at it.

Discussion centered around how to make the task of going from the geodesyml.org site and the resources on github to making an actual implementation simpler. Some ideas discussed included:

Test site with test data
Clone AWS instance with a different agency
Machine-to-machine tests
Starter guide with sample data - how to get up and running and proof of concept (and proof of benefits)

Some additional discussion of ways to structure an implementation. How to get started with GeoServer. How to link to the DB.

Discussion of suitability of MySQL as DB engine. It does have plugins for spatial functionality but not as well-supported as PostgreSQL/PostGIS

4. Revisit our earlier timeline for making progress

We ran out of time to revisit the timeline although it was noted that some of the milestones are essentially complete, and also, under the tasks listed, the initial task of creating the schema from the decision document is done. The use cases are likely to be further developed along the way. The UML diagram is not done but would still be useful to have.

5. Others doing implementation?

Elisabetta D'Anastasio wasn't able to be on the call but reported by e-mail:
"we are going to develop some tool in golang to create geodesyml file for our geonet stations. But we've just started doing that, so I wouldn't have much progress to report"