

CXML Feedback provided by Chris Sisko

(National Hurricane Center / National Weather Service)

The overall impression of the CXML proposal is the power and flexibility that XML brings to limitations inherent in the current data formats (binary or ascii text) utilized today. As outlined in your report, XML is self-describing and scalable. The proposal for CXML has merit and if the format specification is clearly and extensively documented, I believe that CXML would be widely adopted and could be provided in real-time for external use.

Positives of CXML:

- Self describing and easy to use as specified in the report.
- Scalable in such a way that end users are shielded from changes - end user impact is minimized since their applications are less likely to crash (not true with changes made to today's data formats).
- Ability to convey more information about more complex guidance – ensembles, deterministic models, multi-model consensus, interpolated models/tracks, etc.
- Potential to be OGC compliant when ratified.
- Ability to embed meta information into the format in order to communicate data limitations and other information about the data.
- Excellent CXML “data model” in the proposal that depicts the one to one, one to many and many to many relationships – very nice indeed.
- Web friendly and SOA compatible (SOAP, etc).

Challenges for CXML:

- Compatibility with assistive web technologies? Can CXML work with assistive technologies, especially in countries that require equal access for all? In the US, the requirement to be compatible with assistive technologies is called “508 compliance” and this can be an issue.
- Conveying “manual” or “official” forecast information is difficult because it involves watches/warning areas and depiction of hazards (rainfall, flooding, tornadoes, storm surge, mud slides, etc). How will CXML be able to convey these critical elements of an official forecast?
- CXML has an excellent ability to store model guidance information. However, over a tropical cyclone's life cycle the bulk of information will be extremely large since it will contain ensembles, deterministic models, multi-model consensus output, interpolated models, etc. How best can the bulk of data be handled? Are the only options to use compression or breaking the data into fragments? What are the pros and cons of doing either?
- Will CXML leverage CAP (Common Alert Protocol) to convey emergency information to provide greater interoperability with other systems used by emergency managers, defense, first responders, etc?
- The proposal document was unclear in differentiating TC fix/analysis information versus TC final best track information. NHC, CPHC and JTWC (US Navy) define “best track” as the final approved track, intensity, etc of a tropical cyclone based on final analysis of all

available information. Is there a clearer way to define “best track” information in the CXML format so the end user can clearly identify this information as the “final” analysis?

- Standardization of IDs, especially unique IDs is a challenge, especially when systems cross into other areas of responsibility. Although CXML has the flexibility to use “local IDs”, it appears the primary ID `<disturbance ID="2007072518_134N_1102E">` is so unique that it is valid for only that instance in time and space. It may be very difficult to merge disparate data from other centers who may be working the same system unless standard IDs are used. Are standard unique IDs being explored?
- Can meta information (general descriptions about the data) be formally established in CXML similar to what is done in GIS shapefiles? Would this meta information be part of the header or some expansion to the header information if included?
- Expansion of the Fix information like the Dvorak information is clearly needed. Current CXML specification contains a hard to decipher string:

`<DvorakIntensityCode>T3.0/3.0/D1.0/24HRS</DvorakIntensityCode>` This can be described better in CXML convention by defining extra tags and adhering more closely to XML common practices:

```
<DvorakIntensityCode><RawTNum>T3.0</rawTNum><CINum>3.0</CINum>
<DevelTrend>D</DevelTrend><DevelAmtChange>1.0</DevelAmtChange><AvgPd>24HRS</AvgPd>
</DvorakIntensityCode>
```

Recommendations:

- It may be worthwhile for the CXML working group to invite more members, especially RSMCs and other centers that produce large volumes of critical tropical cyclone information to participate.
- Once a final “detailed” CXML specification is drafted, another request for comments should be submitted in order to solidify the format contents.
- The proposed timeline for implementation is too aggressive. A well planned phased rollout approach to disseminating select CXML data (such as guidance or forecast information) first would provide users with more time to produce these datasets without impacting resources and minimize data quality issues.

This feedback only represents the personal viewpoint of Chris Sisko only and does not necessarily represent the view of the National Hurricane Center. Chris Sisko works for the National Hurricane Center as a meteorologist and computer scientist specializing in software engineering. Chris Sisko designs code and maintains NHC’s satellite, data flow, display and ATCF (Tropical Cyclone) systems and as required works hurricane duty as conditions dictate.

Email: chris.a.sisko@noaa.gov

Last update: 20 Feb 2008