

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.

We haven't even considered this. I wouldn't see CXML as an end product, but rather as an exchange format that the people who do value-adding will use to make products that may need to be 508 compliant.

- 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?

The exchange of warning data is a slightly different task than what we have in mind for CXML, which is more the exchange of observation and forecast data. The warnings are really interpretations of the obs and fcsts. When I was looking for XML formats for cyclones, I found a few XML formats that were designed to carry warnings (for example, Renato Ianella at NICTA has done a lot of work in this area in conjunction with OASIS and CAP), but none that were designed to carry much data on the storm characteristics.

- 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?

I suppose it's up to the users, but I would envisage a new CXML file being created whenever a new set of observations became available, or model run was made. For TIGGE in particular, each EPS run from a model would generate a single CXML file that could have more than one tropical cyclone in it. Extracting forecasts for individual storms would probably be a separate process. Many organizations have databases for TC data, and this would be the better way to store the information in the CXML files, rather than keeping an archive of CXML files. It's really meant as an exchange format.

The topic of file names was raised in a meeting earlier this week. We haven't made any recommendations on this, but will have to do so for the TIGGE project.

- 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?

As I understand it, CAP has the ability to reference other XML files, and this is how I'd see the CXML being potentially connected in a direct way. This could work quite well for observations. The model forecasts might require some further post-processing or division into individual storms of interest before attaching to CAPS.

- 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?

This is a good point. There are two elements, `<objTechNum>` and `<objTech>` which came straight out of the ATCF format, and specification of best track data would be given as `<objTech>BEST</objTech>`. I'm not very happy with these tag names – maybe `<objTech>` could become `<objectiveTechnique>` and `<objTechNum>` is no longer needed. In the header section of the file one could also include something like `<product>Best track cyclone analysis</product>`.

- 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?

We've been talking about it a lot lately but have not come up with any better ideas. For observations there could be a chance if the TC data providers could agree on at least some convention, but for NWP forecasts it's hard to imagine how the names would be assigned. What we're mainly after is some way to associate the same storm in consecutive files/forecasts from a single producer. The ID that we have suggested gives time and space information corresponding to the first detection and may be more useful in merging data from multiple centers than the local IDs. However, it's probably not reliable enough to be used with confidence and people will still have to use intelligent merging schemes. I don't see any way around this, at least not until the storm is officially named and this name is included in the file, e.g.

```
<cycloneName>George</cycloneName>.
```

- 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?

We thought about using XLink to do this, but didn't end up going that way. The header can point to metadata living elsewhere by providing URLs in either (or both) `<moreInfo>` or `<moreMetadata>`, but there's no guidelines on what those URLs would link to.

- 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>
```

Good point. This suggestion was also made by a few others and has been put into the development version of CXML.

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. Agreed! We wanted to develop a prototype and get some feedback on it – that's where we're at now. Now that we've seen that the TC community is behind the idea, it would be a good time to invite some more experts to contribute to its development.

- Once a final "detailed" CXML specification is drafted, another request for comments should be submitted in order to solidify the format contents.

I think we're getting close to a final agreement on the data contents. A detailed spec has not yet been written but that's the next step.

- 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.

We have been rushing, I have to admit. The T-PARC experiment was one of the drivers for this work, and it is due to begin later this year. This will be seen as a test for the use of CXML for NWP forecasts, and will no doubt suggest improvements for future versions of CXML. In general, though, I agree that a more planned rollout would be more sensible.