**CoT-NIEM-0.8-iepd (“Punch List”)  
Release Notes**

1. I had to rework the location types. You can’t extend an adapter type, so I had to take the error estimate out of mof:LocationPoint. Augmentations don’t work the way I thought/hoped, so I needed to put them into a single mof:LocationAugmentation element, which would give us this:

<mof:LocationPoint>

<gml:Point gml:id="I1" srsName="http://metadata.ces.mil/mdr/ns/GSIP/crs/WGS84E\_3D">

<gml:pos>32.23426 -110.34511 90.2</gml:pos>

</gml:Point>

</mof:LocationPoint>

<mof:LocationAugmentation>

<mof:LocationErrorCylinder>

<mof:LocationCircularErrorRadiusValue>44.2</mof:LocationCircularErrorRadiusValue>

<mof:LocationLinearErrorHalfHeightValue>21.7</mof:LocationLinearErrorHalfHeightValue>

</mof:LocationErrorCylinder>

<mof:LocationCreationCode>MACHINE.GPS</mof:LocationCreationCode>

</mof:LocationAugmentation>

Except mof:LocationCreationCode is supposed to modify the lat/long/hae, not the entire Location. Rats. OK, I can do that, but now I need an element that represents the whole CoT cylinder. Presto! new element mof:LocationCylinder, which gives us this:

<mof:LocationCylinder>

<mof:LocationPoint>

<gml:Point gml:id="I1" srsName="http://metadata.ces.mil/mdr/ns/GSIP/crs/WGS84E\_3D">

<gml:pos>32.23426 -110.34511 90.2</gml:pos>

</gml:Point>

</mof:LocationPoint>

<mof:LocationCylinderRadiusValue>44.2</mof:LocationCircularErrorRadiusValue>

<mof:LocationCylinderHalfHeightValue>21.7</mof:LocationLinearErrorHalfHeightValue>

<mof:LocationCreationCode>MACHINE.GPS</mof:LocationCreationCode>

</mof:LocationCylinder>

1. I discovered Event and EventType in the existing NIEM MilOps domain. Don’t know how I managed to overlook them before. We can’t just ignore those components. Our CoT-NIEM IEPD and Lizzie’s emergency management IEPDs are built on our “milops-future” reference schema. That reference schema contains components I think will be added to NIEM MilOps in the future. So the EventType in milops-future needs to line up with the existing EventType in NIEM MilOps 1.0.

As it turns out, that only requires one change: mof:EventTypeCodeAbstract becomes mof:EventCategoryAbstract. The messages don’t change, you’ll still see cot:EventTypeCode in the instance data.

By the way, if someone says “forget all that future stuff, build me a CoT IEPD based on the NIEM content that exists today”, that’s easy to do. We would just move a bunch of components from milops-future into cot-niem.xsd. The amount of reuse / commonality between CoT-NIEM and Lizzie’s other IEPDs would be a lot smaller, of course.

1. OK, now for the punch list. First agreed change: LocationCreationCode should modify LocationPoint. Handled as described in #1 above.
2. Reordered mof:EventType elements so that mandatory elements appear first
3. Made nc:CommentText a mandatory field in nc:CommentType subset
4. Removed cot:EventAugmentationPoint
5. Removed mof:EventIDAbstract; just have EventID
6. Moved the EventContact element to milops-future, and used nc:ContactInformationType
7. That’s it for the punch list. I also provided documentation for most of the components. Including documentation for those pesky roll, pitch, and yaw properties. See what you think.
8. And I ran the schemas through the NIEM 3.0 conformance rules. This flagged a few components for breaking the naming rules. Fixed:
   1. @mof:sigmaError becomes sigmaErrorValue
   2. @mof:codespace becomes codespaceID
   3. mof:CodespaceCodeType must extend niem-xs:token, not xs:token
9. And I changed some properties in EventLink to provide better semantics.
10. And I defined components for the “contact” subschema.
11. And I wrote embedded Schematron assertions for the GML components in milops-future.
12. And I created some needed IEPD artifacts: mpd-catalog.xml, master-document.txt, and changelog.txt
13. And I updated the GML subset schema in xsd-exi.
14. And I reworked the OWL taxonomies for MIL-STD2525 and CoT event type code.

I don’t think there’s much more to be done.