**NIEM-CoT-0.6-iepd (“Probably Good Enough”)  
Release Notes**

This release adds:

1. EventTypeCode enumerations
2. UID subschema
3. Link subschema
4. Flowtag subschema

EventTypeCode enumerations are provided in extension/codes-enum.xsd. The canonical list of CoT type codes came from types\_milstd\_appendix.docx. The ATOM subtree was built by matching CoT type codes to the function ID codes in MIL-STD-2525C and then using the corresponding 2525C hierarchy code (substituting ATOMS for WAR).

The correlations between CoT type codes and NIEM-CoT codes are given in correlate-edit.xlsx. This is not part of the IEPD, but may be helpful when writing the translator.

I also created OWL files for the EventTypeCode taxonomy and the MIL-STD-2525C taxonomy. These are provided, but not used in the IEPD.

I handled the UID subschema by giving the EventID element a systemID attribute and making it repeatable. By convention the first element corresponds to the CoT core uid attribute and will not have a systemID attribute. The following elements correspond to attribute name/value pairs from the uid subschema.

I handled the Link subschema by defining a new optional, repeatable EventLink element.

I handled the Flowtag subschema by defining a new optional, repeatable EventReportFlowtagDateTime element.

I still don’t have a list of the canonical SensorTypeCode values. There’s nothing in the subset schema or in types.txt. So at present any code will validate. Following Ernie’s suggestion, I also added a SensorTypeExtendedCode element, similar to EventTypeExtendedCode.

This is not a complete NIEM 3.0 IEPD. It is missing most of the required documentation elements, and I haven’t built a mpd-catalog file, etc. It should be good enough for the translator. There is reason to believe that future work on the IEPD will not result in substantial rework on the translator.

I’m happy to change the order of elements, name of elements, etc. Just let me know.

Here are some suggestions for the translator:

1. When translating the CoT type attribute to ncot:EventTypeCode, you should choose the longest match in the correlation table. For example, given “a-.-A-C-F-x-y” you should match “a‑.‑A‑C‑F”, with corresponding code ATOM.AIRTRK.CVL.FIXD. If you can’t match the whole type attribute string, then put the entire string into ncot:EventTypeExtendedCode with no codespace attribute.
2. When translating NCOT to CoT, if you find an EventTypeExtendedCode element with no codespace attribute, use that for the type attribute
3. When translating NCOT to CoT, if you find an EventTypeExtendedCode element that has codespace attribute, write the whole element into the detail element, like this

<detail>

<ncot:EventTypeExtendedCode xmlns:ncot=”http://example.com/niemcot/0.6/”

ncot:codespace=”http://example.com/SillyAnimals”>

ANIMAL.ELIPHANT

</ncot:EventTypeExtendedCode>

</detail>

1. When translating CoT to NCOT, if you find an EventTypeExtendedCode element in the detail element, put it into the output in the right place

I believe rules 1-4 will give us round-trip translation between CoT type attribute and the NCOT elements. But don’t take my word for it.

I also think the same trick will work for SensorExtendedTypeCode, etc.