**A Guided Tour of the CoT-NIEM IEPD**

1. Start with master-document.txt, which explains the purpose of the CoT-NIEM information exchange package description (IEPD).
2. Next look at the sample XML documents in iep-samples. cot-iep1.xml is a CoT-NIEM information exchange package (IEP) that exercises all of the components. cot-2.0-msg.xml is the equivalent information in CoT 2.0 syntax.
3. Now look at extension/milops-future.xsd. This is a subset of a hypothetical future MilOps domain reference schema. There are about 60 new data components. With these you have the basis of a generic what-where-when event message, which can be customized into a CoT equivalent or into something else. Look at iep-samples/milops-event.xml to see a generic MilOps event message.
4. Finally, look at extension/cot-niem.xsd. This is an extension schema defining CoT-NIEM data components which we do not hope or expect to find in the future MilOps domain.
5. CoT-NIEM IEPD statistics:
   1. Components reused from NIEM Core 69
   2. Components reused from MilOps-Future 58
   3. Components created specially for CoT-NIEM 79
   4. Extensions of core components 5
   5. Substitutions for core components 11

Other things worth observing:

1. Values for cot:EventTypeCode are drawn from a taxonomy described in extension/event-type-code.owl. The classes in that OWL taxonomy are mapped to the classes in extension/milstd-2525C-sym.owl. This illustrates what is needed to use OWL taxonomies in NIEM messages.
2. The business rules for CoT-NIEM IEPs in schematron/business-rules.sch. These define additional constraints that would be difficult or impossible to define in XML Schema.
3. The schematron business rules attached to data components within a reference schema in milops-future.xsd. The missing context for each rule (sch:pattern, sch:rule, etc.) can be automatically generated from the reference schema. To see what that completed context would look like, examine schematron/schema-rules.sch
4. The way that cot-niem.xsd first restricts, then extends the generic EventType in milops-future. See how CoT-NIEM imposes its own (more restrictive) EventID and EventTypeCode.
5. The use of the xs:any wildcard in cot:EventCommunityExtensions. This is allowed by the NDR and is essential to lossless round-trip translation. It does require making a subset of structures.xsd for the IEPD, which also is allowed (but not presently supported by SSGT).
6. The customized schemas for EXI encoding, xsd-exi/xs.xsd and xsd-exi/gml.xsd. These permit more efficient encoding than the stock schemas used for validation and conformance. The file xml-catalog-exi.xml defines the exact schema set to be used when EXI encoding and decoding a CoT-NIEM IEP.

In the future:

1. I’d like to see SSGT modified so that everything it generates falls into the niem directory. That would require moving wantlist.xml and xml-catalog.xml into niem. I’d also like a convention which says that everything in the niem directory is SSGT output (presuming you are using SSGT). If you want to manually muck around with a subset schema, then you copy it into another directory and adjust the catalog files. I’ve done all that rearranging in this IEPD, so you can see what it would look like.