This is a really excellent review of this issue. It is thorough and thoughtful. In addition, it provides some keen observations and helpful recommendations. We’ve provided some additional thoughts and observations below using Word revision markings.

# Issue.

USMTF has a requirement to maintain the sequence, or order, of repeatable fields in a set format to support slash-delimited message text format (MTF) representations. In XML-MTF, this is accomplished by wrapping the repeatable fields inside a locally declared <GroupOfFields> element that has its own cardinality (minOccurs, maxOccurs). Local element declarations are prohibited by the following NIEM rule for Reference (REF) and Extension (EXT) schema:

**Rule 9-35. Element declaration is top-level**

**[Rule 9-35] (REF, EXT) (Constraint)**

<sch:pattern>

<sch:rule context="xs:element[exists(@name)]">

<sch:assert test="exists(parent::xs:schema)"

>An element declaration MUST be top-level.</sch:assert>

</sch:rule>

</sch:pattern>

The purpose of this paper is to present options for making a decision on the way ahead for USMTF NIEM transition of the <GroupOfFields> element.

1. **Discussion.**

# MIL-STD-6040 Group of Fields (aka Field Group) definition and schema declaration.

Group of Field Formats as they apply to USMTF occurs in two ways, columnar and linear. Columnar Sets can be identified because the set identifier will always start with a numeric (e.g., 1MSNLOC). A columnar set is a formatted set having an ordered collection of data aligned vertically under a horizontal array of column headers and all fields in the set will be repeatable. Linear Sets have a collection of field formats that must be repeated in the order of occurrence. In this case the repeatable fields must occur at the end of a set in order to maintain backward compatibility with the slash delimited structure.

MIL-STD-6040 provides the following guidance on field repetition.

**5.8.9 Field Repetition**. There is often a need for repetition of certain types of information within a set. To support this need, one or more fields of a set may be repeated as described in the subparagraphs below, but only when so designated by the set format. Normally, there is no limit to the number of repetitions; however, the number of repetitions may be unconditionally limited by specification in the set format specification. Additionally, the number of repetitions may be conditionally limited, in accordance with expressions described in Appendix C. There is no restriction on the selection of an alternative in repeated alternative content fields unless that restriction is accomplished in accordance with expressions described in Appendix C.

1. **Repetition Within Linear Set**. In the linear set, the designated final field, or group of contiguous fields that includes the final field, may be repeated. Repetition of the final fields of a set as a group of fields must be accomplished in a way that allows context meaning of repeated field positions to be maintained. Therefore, when a group of fields is designated for repetition, it must be repeated as a unit. In repeating the final fields of a set as a group of fields, the order of fields within the group must be maintained, even if some of the data were not available or being withheld.
2. **Repetition Within Columnar Set**. All the fields of a columnar set comprise a repeatable group of fields as explained in paragraph 5.8.4. In columnar sets, each instance of the group of fields is always entered on a new line in order to maintain the desired tabular arrangement.

**E.8.8 Group of Fields Mapping**. A Group of Fields is defined as collection of Fields in a Linear Set that is associated for repetition and is located at the end of a Set Format. Columnar Set Formats are comprised only of a Group of Fields. Each occurrence of an MTF Group of Fields is mapped to an XML-MTF group-of-fields element. The Column Headers of Columnar Sets are not mapped. The element has the following structure:

**<GroupOfFields>**(FIELD)+**</GroupOfFields>**

The content of an XML-MTF group-of-fields is mapped from the MTF Fields contained in the MTF Group of Fields, and is composed of one or more XML-MTF field elements.

Example:

JAM/UTMK:32WDL1212/141325Z/5MIN/LL12345/32 RCS/HTH/4.5/4/33/none/LL12346/32RCS/EW/3.4/1/0/NSTR//

The underlined portion of the above MTF Jamming Encountered Set maps to the following in XML-MTF:

<GroupOfFields>

<MissionNumber>LL12345</MissionNumber>

<TaskedUnit>32 RCS</TaskedUnit>

<AircraftType>HTH</AircraftType>

<DurationTimeInHoursAndTenths>4.5

</DurationTimeInHoursAndTenths>

<CountofSorties>4</CountOfSorties>

<NumberOfPersonnelRecovered>33</<NumberOfPersonnelRecovered>

<SarMissionReportComment>None</SarMissionReportComment>

</GroupOfFields>

<GroupOfFields>

<MissionNumber>LL12346</MissionNumber>

<TaskedUnit>32 RCS</TaskedUnit>

<AircraftType>EW</AircraftType>

<DurationTimeInHoursAndTenths>3.4

</DurationTimeInHoursAndTenths>

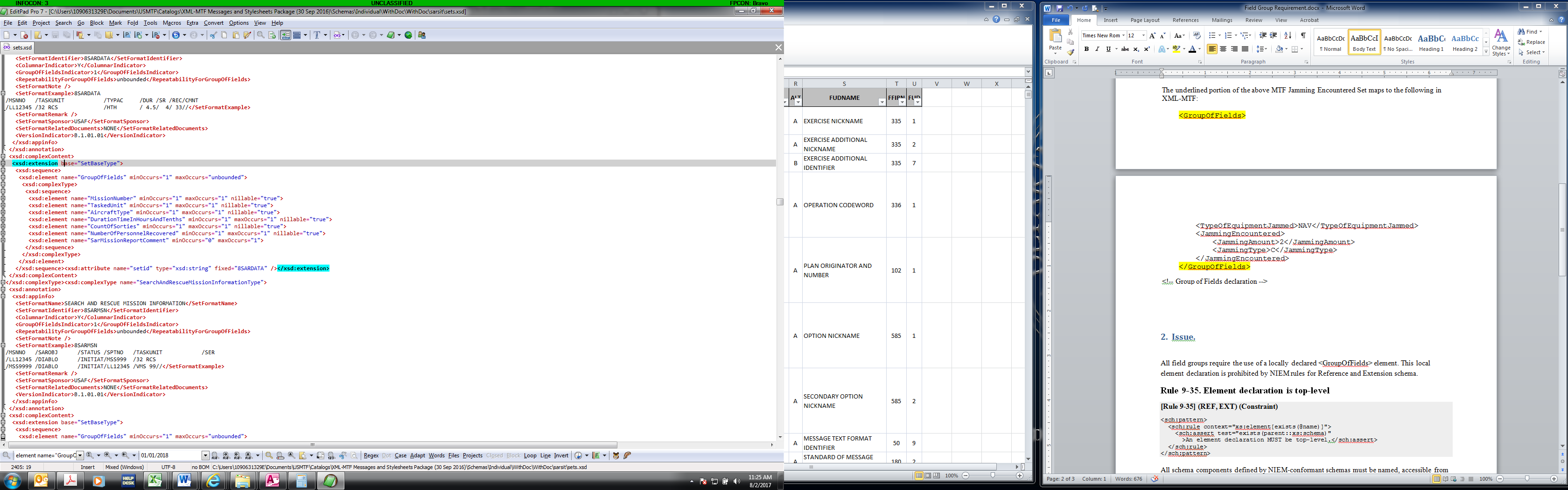
<CountofSorties>1</CountOfSorties>

<NumberOfPersonnelRecovered>0</<NumberOfPersonnelRecovered>

<SarMissionReportComment>NSTR</SarMissionReportComment>

</GroupOfFields>

<!—Example of a Group of Fields .xsd declaration -->



# All schema components defined by NIEM-conformant schemas must be named, accessible from outside the defining schema, and reusable across schemas requiring named NIEM elements to be top level which ensures they are globally reusable.

1. In the example above the <GroupOfFields> element is declared locally. If Sets are to be represented as either Reference or Extension schema conformant, each locally declared <GroupOfFields> element will have to be replaced by a unique, globally declared element which has a corresponding globally declared complexType.

# Impact.

There are 808 sets with field groups in the USMTF Data Base. Of these, 256 are columnar set field groups and 552 are linear set field groups. Of the linear sets with field groups, 296 have a single repeatable field. In the case of single repeatability fields, we are recommending they not be contained inside a <GroupOfFields> element as required by MIL-STD-6040. Since we are talking about a single field, there is no sequence to be maintain so nothing is gained by including them inside a <GroupOfFields> element. If this recommendation is agreed that leaves 512 sets that need to be addressed. There are likely some field groups that are the same but have different names but that number will not significantly reduce the number of changes required. We have not done an analysis to determine how many or which messages will be affected.

Agree. This is a helpful observation and recommendation. Removing the unnecessary <GroupOfFields> elements will require some code changes; however, the changes require little effort (i.e., deleting unnecessary statements) and leave the remaining code smaller and simpler.

In fact, the idea of collapsing unnecessary <GroupOfFields> elements when only *one* of the fields in a set repeats could also be extended to cases where *all* of the fields in a set repeat. This can be accomplished using the same approach used when one field repeats. I.e., remove the repeating <GroupOfFields> element and make the parent element repeatable instead. For example, the following XML representation for the 8LIFTREQ set:

<AirliftRequestInformation>  
 <GroupOfFields>  
 <SecurityClassification>U</SecurityClassification>  
 <RequestNumber>001</RequestNumber>  
 <Priority>  
 <AirliftRequestPriority>2B2</AirliftRequestPriority>  
 </Priority>  
 <AirliftMissionType>SAAM</AirliftMissionType>  
 <RequestingUnit>101ST ABN</RequestingUnit>  
 <AirliftComments>S1, DEPLOY</AirliftComments>  
 </GroupOfFields>  
 <GroupOfFields>  
 <SecurityClassification>U</SecurityClassification>  
 <RequestNumber>002</RequestNumber>  
 <Priority>  
 <AirliftRequestPriority>2B21B</AirliftRequestPriority>  
 </Priority>  
 <AirliftMissionType>PER</AirliftMissionType>  
 <RequestingUnit>101ST ABN</RequestingUnit>  
 <CargoDeliveryMethod>PER</CargoDeliveryMethod>  
 <AirliftComments>S2, DEPLOY</AirliftComments>  
 </GroupOfFields>  
 </AirliftRequestInformation>

could be modified to have the following representation:

<AirliftRequestInformation>  
 <SecurityClassification>U</SecurityClassification>  
 <RequestNumber>001</RequestNumber>  
 <Priority>  
 <AirliftRequestPriority>2B2</AirliftRequestPriority>  
 </Priority>  
 <AirliftMissionType>SAAM</AirliftMissionType>  
 <RequestingUnit>101ST ABN</RequestingUnit>  
 <AirliftComments>S1, DEPLOY</AirliftComments>  
 </AirliftRequestInformation>

<AirliftRequestInformation>  
 <SecurityClassification>U</SecurityClassification>  
 <RequestNumber>002</RequestNumber>  
 <Priority>  
 <AirliftRequestPriority>2B21B</AirliftRequestPriority>  
 </Priority>  
 <AirliftMissionType>PER</AirliftMissionType>  
 <RequestingUnit>101ST ABN</RequestingUnit>  
 <CargoDeliveryMethod>PER</CargoDeliveryMethod>  
 <AirliftComments>S2, DEPLOY</AirliftComments>  
 </AirliftRequestInformation>

This technique can be applied in cases where all the fields in a set are repeatable, as long as the set itself is not also repeatable. This is true for all columnar sets. Appling this technique to all columnar sets would eliminate another 256 instances of the <GroupOfFields> element, leaving 256 to be addressed.

It is also true for 85 of the 91 linear sets in which all fields are repeatable. Appling this technique to these 85 sets would leave only 171 instances of the <GroupOfFields> element.

There are only 6 linear sets in which all fields are repeatable, but this technique cannot be applied because the set itself is repeatable. They are NEWDTSUM, STATION, ROEAPPR, ROEIMPL, ROEREQ and ROEPRCX. On review, it is not yet clear why both the set and all the fields in the set repeat in these instances. If these two levels of repetition are redundant and accomplish largely the same thing, it is possible one of them might be removed. Given the small number of sets where this occurs, we recommend these sets be reviewed individually and handled on a case-by-case basis.

# Solution.

# There are several options that can be applied to this situation based on the desired outcome. Below we will discuss each option.

1. Option 1. Make no changes. This means we keep the <GroupOfFields> element locally declared. If this option is selected, Sets cannot go into NIEM SSGT which limits the visibility of our Set data to the NIEM users.

Note: This would also mean these sets would not be NIEM compliant since NIEM rule 9-35 is required for both REF and EXT schemas.

1. Option 2. Design REF schemas for submission to NIEM for the 1422 linear sets that do not include two or more repeatable fields. Keep the current structure for the 512 sets with two or more repeatable fields. This option would require two Set structure formats in the schema generation tool and it assumes MIL-STD-6040 rules would be changed to exclude single repeatable fields from the <GroupOfFields> concept.

Note: As with the case above, this option would mean these 512 sets with two or more repeatable fields would not be NIEM compliant.

1. Option 3. Change NIEM rule 9-35 from MUST to SHOULD or make exception for USMTF <GroupOfFields> element since this element has a specific purpose unique to USMTF.

We recommend pursuing this option. We believe this would be a general improvement to NIEM that would further reduce barriers to adoption for a variety of existing systems – not just USMTF. In addition, it would help to address over 1,000 additional USMTF name changes due to Field names that conflict with other Field and FUD names (see USAF analysis from action item 2017-2-2). Not every element in an XML schema represents a reusable, semantically independent concept. Some elements are designed for a very specific context and some are purely structural, representing parts of a whole that may be omitted or repeated. The <GroupOfFields> construct in USMTF is a great example of this as are USMTF alternate content fields and many USMTF segments.

The NIEM philosophy of maximizing reuse of data definitions is a very good one. However, requiring every single element to be global and reusable – even if they are context dependent and have no independent semantic value -- is a relatively extreme policy. Not all elements are well suited for reuse. And as we’ve seen in our testing, making all elements global has negative consequences for both system developers and end users. Schema designers should be heavily encouraged to identify and expose potentially reusable components, but should not be forced to expose every element globally. As such, we recommend this rule be relaxed.

Note also that global schema types provide most of the benefits of global elements without the negative consequences. So, relaxing this rule and keeping rules 9-10 and 9-24 (global simple types and complex types), would still provide most of the value without the consequences.

1. Option 4. Replace locally declared <GroupOfFields> element with unique, globally declared field group elements with associated globally declared field group complexTypes. This would require a redesign of the schema and the creation of as many as 512 unique names for the globally declared field groups.
2. Option 5. Add cardinality (minOccurs=”(0|1)” maxOccurs=”(1|maxrepeat|unbounded)” to the xs:sequence element and eliminate the <GroupOfFields> element altogether. This option is prohibited by NIEM rules for REF, EXT schema and would require a change or exception to the following NIEM rules:

At first glance, this looks like a pretty attractive solution as well. However, we ran some tests and found this method introduces some of the same implementation complexities as substitution groups. In particular, widely used binding layers like JAXB return the contents of the xs:sequence as a List of generic objects that conceal available help text, defeat code-completion and require longer, more complex code for dynamically determining the type of each element. Therefore, we do not recommend this approach.

**9.3.2.1. Sequence cardinality**

XML Schema allows each particle to specify cardinality (how many times the particle may appear in an instance). NIEM restricts the cardinality of xs:sequence particles to exactly one, to ensure that content model definitions are defined in as straightforward a manner as possible.

A schema developer who requires the instance syntax that would be obtained from the use of specific cardinality on sequences should define cardinality on individual element uses.

**Rule 9-65. Sequence has minimum cardinality 1**

**[Rule 9-65] (REF, EXT) (Constraint)**

<sch:pattern>

<sch:rule context="xs:sequence">

<sch:assert test="empty(@minOccurs) or xs:integer(@minOccurs) = 1"

>An element xs:sequence MUST either not have the attribute {}minOccurs, or that attribute MUST have a value of 1.</sch:assert>

</sch:rule>

</sch:pattern>

**Rule 9-66. Sequence has maximum cardinality 1**

**[Rule 9-66] (REF, EXT) (Constraint)**

<sch:pattern>

<sch:rule context="xs:sequence">

<sch:assert test="empty(@maxOccurs) or (@maxOccurs instance of xs:integer

and 1 = xs:integer(@maxOccurs))"

>An element xs:sequence MUST either not have the attribute {}maxOccurs, or that attribute MUST have a value of 1.</sch:assert>

</sch:rule>

</sch:pattern>

1. Option 6. TBD