**ECF5 Spec Feedback and Considerations – 3**

JEC responses in red

This document contains additional questions and commentary resulting from a review at the Electronic Court Filing Version 5.0 Working Draft 08.

1. **FiducuaryTypeCode**

FiducuaryTypeCode is described as:

“Legal description of the role of a fiduciary. This is needed in addition to the Case Role Code (in the Review Filing message structure) because that code only defines the roles of other actors and other parties; the fiduciary per specification is the initiating party. Examples: guardian, trustee, conservator of the person, conservator of the estate.” (underline added)

Since this is “per specification”, then where in the specification is this?

It is presumed that ‘Case Role Code’ refers to CaseParticipantRoleCode.

So what exactly does the FiduciaryTypeCode description mean or imply?

Firstly, this definition suggests that ecf:CaseParticipantRoleCode should be used:

<ecf:CaseParticipantRoleCode>Fiduciary</ecf:CaseParticipantRoleCode>

Secondly, the description implies that above is insufficient, therefore, since the fiduciary is the initiating party, this description implies that j:CaseInitiatingParty should also be used.

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:id="Person1">

 <nc:PersonName>

 <nc:PersonGivenName>John</nc:PersonGivenName>

 <nc:PersonMiddleName>W.</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 </nc:PersonName>

 <ecf:PersonAugmentation>

 <ecf:CaseParticipantRoleCode>Fiduciary</ecf:CaseParticipantRoleCode>

 <ecf:FilerIdentification>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:FilerIdentification>

 </ecf:PersonAugmentation>

 </nc:EntityPerson>

</j:CaseInitiatingParty>

In Arizona, our SMEs say that a fiduciary is never a party in a case, much less as always the initiating party.

I propose we change the definition to:

Legal description of the role of a fiduciary. Examples: guardian, trustee, conservator of the person, conservator of the estate.

1. **Type of Party**

ECF 5 provides elements for the following three types of case parties:

Element Description

J:CaseDefendantParty An entity being charged or sued in a court of law.

J:CaseInitiatingParty An entity that brings charges or a suit against another in a court of law. Can be either a victim in a criminal case or a plaintiff in a civil case.

J:CaseRespondentParty An entity in a court case that is required to answer a petition for a court order or writ requiring the respondent to take some action, halt an activity or obey a court’s direction. In such matters the moving party (the one filing the petition) is usually called the petitioner. Thus, the respondent is equivalent to a defendant in a lawsuit. On an appeal, the party who must respond to an appeal by the losing party in the trial court (called appellant) in the appeals court. The accused in a domestic violence case or civil action; a person responding to a Request or Petition for Protection filed by a petitioner.

These role-specific party elements seem to be too few to express all the litigant CaseParticipantRoleCode types suggested in the ‘combined’ CaseParticipantRoleCode.xlsx. If these three types truly are sufficient, then every litigant type code in the spreadsheet should be able to be mapped to one and only one of the three elements listed above. In which element would ‘Intervenor’ be mapped? How about a cross-claimant? RealPartyInInterest? CounterPlaintiff?

NIEM 3.2 and ECF 5.0 also include j:CaseOtherEntity for case parties which, combined with CaseParticipantRoleCode, should be sufficient.

1. **CaseParticipant AWOL**

The CaseParticipant element is no more!

My initial reaction is that this is alarming!

Per the mapping spreadsheet (i.e. model/niem-mapping.csv), nc:CaseParticipant has been replaced by nc:Case/j:CaseAugmentation/j:CaseOtherEntity.

Unfortunately this is not an equivalent or satisfactory replacement.

j:CaseOtherEntity (ECF5) only provides:

 structures:ObjectAugmentationPoint

 nc:EntityRepresentation ( nc:EntityOrganization or nc:EntityPerson)

nc:CaseParticipant (ECF4) provided:

 nc:EntityRepresentation( EntityItem, EntityItemReference, EntityOrganization, EntityOrganizationReference, EnityPerson, or EntityPersonReference)

 ecf:CaseParticipantRoleCode

 nc:ContactInformation

ecf:CaseParticipantRoleCode and nc:ContactInformation are now included in ecf:PersonAugmentation and ecf:OrganizationAugmentation.

Since in ECF5, EntityPerson and EntityOrganization can provide both structures:id and structures:ref, EntityPersonReference and EntityOrganizationReference may no longer be needed. We do however still need EntityItem (and an equivalent for EntityItemReference).

NIEM 3.2 does not include EntityItem or any elements ending in Reference. I proposed that we can model property as an Organization. Otherwise, if the TC decides we need an EntityItem, it will need to be an extension. TODO: Discuss with the TC.

Also, ecf:CaseParticipantRoleCode has now been relocated. It appears in:

* EntityPerson/ecf:PersonAugmentation
* EntityOrganization/ecf:OrganizationAugmentation
* nc:RoleOfPerson/ecf:PersonAugmentation
* j:CaseOfficial/ecf:CaseOfficialAugmentation

I’m not sure if ecf:CaseParticipantCode appears within EntityItem, but it needs to (just as EntityItem needs to appear in nc:EntityRepresentation).

It is not clear what benefit is obtained by retiring CaseParticipant in favor of j:CaseOtherEntity.

There is a key NIEM principle that extensions should only be created when there is not a comparable NIEM element. The UML definition of CaseParticipant is “Placeholder for all case participants, i.e. persons or organizations playing a role in a case. This is where all actor details are expressed.” The NIEM definition of CaseOtherEntity is “A miscellaneous entity involved in a court case.” I propose these are sufficiently similar. Otherwise, if the TC decides we need a CaseParticipant element, it will need to be an extension. TODO: Discuss with the TC.

1. **ID/IDREF Reference Hell**

Also, nc:EntityRepresenation in ECF5 now only provides EntityOrganization or EntityPerson. In ECF4, nc:EntityRepresentation provides EntityItem, EntityItemReference, EntityOrganization, EntityOrganizationReference, EntityPerson, and EntityPersonReference.

In ECF4 EnityPerson and EntityOrganization have an s:id attribute, but they do not have an s:ref attribute. With ECF5, EntityPerson and EntityOrganization have both structures:id and structures:ref attributes. As such, EntityPersonReference and EntityOrganizationReference may not be needed any longer.

However, what prevents misuse?

In fact, illogical or nonsensical things are now possible in ECF5 that were not possible in ECF4. For example:

In ECF4, when the person who was submitting the filing was also the initiating party (which is not always the case), then the following CoreFilingMessage could be used to communicate this as shown in the following XML fragment:

<CoreFilingMessage …

 <nc:DocumentSubmitter>

 <ecf:EntityPersonReference s:ref="Person1"/>

 </nc:DocumentSubmitter>

 …

<cc:CivilCase …

 <j:CaseInitiatingParty>

 <nc:EntityPerson s:id="Person1">

 <nc:PersonName>

 <nc:PersonGivenName>John</nc:PersonGivenName>

<nc:PersonMiddleName>W.</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 </nc:PersonName>

 </nc:EntityPerson>

 </j:CaseInitiatingParty>

The DocumentSubmitter person is understood to be the same person as the CaseInitiatingPartyPerson. The corresponding ECF5 FilingMessage would appear as:

<filing:FilingMessage …

 <nc:DocumentSubmitter>

 <nc:EntityPerson structures:ref="Person1"/>

 </nc:DocumentSubmitter>

 …

 <nc:Case …

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:id="Person1">

 <nc:PersonName>

 <nc:PersonGivenName>John</nc:PersonGivenName>

 <nc:PersonMiddleName>W.</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 </nc:PersonName>

 </nc:EntityPerson>

 </j:CaseInitiatingParty>

So far so good. However, the ECF5 example can be manipulated to provide nonsensical/illogical constructs that are not possible in ECF4, such as:

For starters, in ECF5 you can state:

 <nc:DocumentSubmitter>

 <nc:EntityPerson structures:id="Person10" structures:ref="Person10"/>

</nc:DocumentSubmitter>

Of course this is nonsense, but it is valid per schema. This type of nonsense is not possible in ECF4.

With ECF5, you could also provide:

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:id="Person1" structures:ref="Person2">

 <nc:PersonName>

 <nc:PersonGivenName>John</nc:PersonGivenName>

 <nc:PersonMiddleName>W.</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 </nc:PersonName>

 <ecf:PersonAugmentation>

 <ecf:CaseParticipantRoleCode>Defendant</ecf:CaseParticipantRoleCode>

 <ecf:FilingPartyID>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:FilingPartyID>

 </ecf:PersonAugmentation>

 </nc:EntityPerson>

 </j:CaseInitiatingParty>

 <j:CaseRespondentParty structures:id="Person2">

 <nc:EntityPerson>

 <nc:PersonName>

 <nc:PersonGivenName>Jane</nc:PersonGivenName>

 <nc:PersonMiddleName>Q</nc:PersonMiddleName>

 <nc:PersonSurName>Doe</nc:PersonSurName>

 <nc:PersonMaidenName>Smith</nc:PersonMaidenName>

 </nc:PersonName>

Of course this too is rubbish, but valid per schema. Again, this nonsense is not possible with ECF4.

And of course, the circular reference possibility is also allowed (and valid per schema) in ECF5 but not with ECF4:

 <j:CaseInitiatingParty>

 <nc:EntityPerson structures:id="Person1" structures:ref="Person2">

 <j:CaseRespondentParty>

 <nc:EntityPerson structures:id="Person2" structures:ref="Person1">

This too is valid per schema with ECF5:

 <nc:PersonAssociation>

 <nc:Person structures:id="Bogus01" structures:ref="Person1" xsi:nil="true"/>

 <nc:Person structures:id="Bogus02" structures:ref="Person2" xsi:nil="true"/>

 <ecf:PersonAssociationAugmentation>

 <ecf:EntityAssociationTypeCode>spouse</ecf:EntityAssociationTypeCode>

 </ecf:PersonAssociationAugmentation>

 </nc:PersonAssociation>

It seems an element should be either an ID type or a REF type but should not be both.

Oh, but wait, …, this appears to be a new NIEM thing. It seems that many, if not all elements have the following set of attributes:



 <xs:complexType name="ObjectType" abstract="true">

 <xs:sequence>

 <xs:element ref="structures:ObjectAugmentationPoint" minOccurs="0" maxOccurs="unbounded"/>

 </xs:sequence>

 <xs:attribute ref="structures:id"/>

 <xs:attribute ref="structures:ref"/>

 <xs:attribute ref="structures:metadata"/>

 <xs:attribute ref="structures:relationshipMetadata"/>

 <xs:anyAttribute namespace="urn:us:gov:ic:ism urn:us:gov:ic:ntk" processContents="lax"/>

 </xs:complexType>

It feels as though NIEM has just driven a very fundamental object orientation principle right into a ditch. How does one interpret the following XML fragment (revised from the civil.xml example)?

 <ecf:CaseAugmentation structures:id="Bogus04" structures:ref="Bogus03">

 <ecf:CaseCategoryCode structures:id="Bogus09" structures:ref="Person2">Estate</ecf:CaseCategoryCode>

 <nc:Metadata structures:id="Metadata1" structures:ref="Bogus02">

 <nc:SensitivityText structures:id="Bogus03" structures:ref="Person1">public</nc:SensitivityText>

 <nc:LanguageCode structures:id="Bogus05" structures:ref="Person2">eng</nc:LanguageCode>

 </nc:Metadata>

 <nc:PersonAssociation structures:id="Bogus06" structures:ref="Person1">

 <nc:Person structures:id="Bogus01" structures:ref="Person1" xsi:nil="true"/>

 <nc:Person structures:id="Bogus02" structures:ref="Person2" xsi:nil="true"/>

 <ecf:PersonAssociationAugmentation structures:id="Bogus08" structures:ref="Bogus07">

 <ecf:EntityAssociationTypeCode structures:id="Bogus07" structures:ref="Person3">spouse</ecf:EntityAssociationTypeCode>

 </ecf:PersonAssociationAugmentation>

 </nc:PersonAssociation>

</ecf:CaseAugmentation>

This is totally unfathomable, but it is valid!

To promote interoperability (much less understanding), we need some rules!

Here are some things to consider:

* An element can use structures:id or structures:ref, but not both.
* Specify that circular references are not permitted (this would also require defining exactly what it means to be circular).
* When using structures:ref attribute, the element containing the destination structures:id attribute must be of the same element type (e.g. element nc:Person could contain a structures:ref that pointed to another nc:Person element or could also point to an nc:RoleOfPerson element since both nc:Person and nc:RoleOfPerson are of type nc:PersonType). Note – if this rule were adopted, then ecf:PayerNameText in PaymentMessage would need to be revised to be nc:PersonType.
* Define in the specification, standardized id/ref relationships and what they mean. For example, when an element of nc:PersonType references another element of nc:PersonType (using structures:ref), then this means the very same person is being referred to..
* Provide guidance on non-specification standardized id/ref relationships.
* Require non-specification standardized id/ref relationships to be defined in court policy.

As you have discovered, NIEM 3.x changed the way references work and we need rules to constraint instances in addition to schemas. Fortunately, these rules are spelled out in the NIEM Naming and Design Rules Section 12.2 Reference Elements.

<https://reference.niem.gov/niem/specification/naming-and-design-rules/3.0/niem-ndr-3.0.html#section_12>

1. **Section 2.1.2 OASIS Universal Business Language**

In the ECF 5 spec, only the cac:AllowanceCharge and cac:Payment elements are mentioned. However in the ECF spec (section 1.3.2) Address (i.e. cac:Address) is also mentioned. Initially one might presume that cac:Address is no longer used in ECF5, but this is not the case. Why no mention of cac:Address?

I added cac:Address to section 2.1.2.

1. **Person Association cardinality**

Why is maxOccurs for nc:Person in nc:PersonAssociation only 2?

Whereas this cardinality may be appropriate for describing a set of twins, it is not very useful for other collections of persons (e.g. the Minnesota Twins).

Note, for nc:OrganizationAssociation, maxOccurs is unbounded for nc:Organization.

I changed the cardinality to 2, unbounded.

1. Mapping

In previous feedback I had suggested that a table should be provided that mapped messages (e.g. filing:FilingMessage) to schema (e.g. filing.xsd).

You replied: This mapping is provided in the table in 4.1 Messages. Each schema links to the appropriate schema.

This may not be working as you had intended.

From table 4.1 I find:

| Providing MDE | Consuming MDE | Operation | Input Message XML element(s) | Output Message XML element |
| --- | --- | --- | --- | --- |
| Court Policy | Filing Assembly | GetPolicy | [policyrequest:GetPolicyRequestMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cpolicyrequest.xsd) | [policyresponse:GetPolicyResponseMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cpolicyresponse.xsd) |
| Court Record | Court Scheduling | AllocateCourtDate | [allocatedate:AllocateCourtDateMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Callocatedate.xsd) | [cbrn:MessageStatus](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cniem%5Cdomains%5Ccbrn%5C3.2%5Ccbrn.xsd) |
| Filing Assembly | GetCase | [caserequest:GetCaseRequestMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Ccaserequest.xsd) | [caseresponse:GetCaseResponseMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Ccaseresponse.xsd) |
| GetCaseList | [caselistrequest:GetCaseListRequestMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Ccaselistrequest.xsd) | [caselistresponse:GetCaseListResponseMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Ccaselistresponse.xsd) |
| GetDocument | [documentrequest:GetDocumentRequestMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cdocumentrequest.xsd) | [documentresponse:GetDocumentResponseMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cdocumentresponse.xsd) |
| GetServiceInformation | [serviceinformationrequest:GetServiceInformationRequestMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cserviceinformationrequest.xsd) | [serviceinformationresponse:GetServiceInformationResponseMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cserviceinformationresponse.xsd) |
| Filing Review | DocumentStampInformation | [stampinformation:DocumentStampInformationMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cstampinformation.xsd) | [cbrn:MessageStatus](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cniem%5Cdomains%5Ccbrn%5C3.2%5Ccbrn.xsd) |
| RecordDocketing | [docket:RecordDocketingMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cdocket.xsd)[payment:PaymentMessage](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cpayment.xsd) (optional) | [cbrn:MessageStatus](file:///C%3A%5CUsers%5CJamesECabral%5COneDrive%5Cxml%5Cecf5%5Cschema%5Cniem%5Cdomains%5Ccbrn%5C3.2%5Ccbrn.xsd) |

Form the above table, I see that the RecordDocketing operation receives docket:RecordDocketingMessage and optionally, payment:RecordPaymentMessage.

If you hover the mouse pointer over docket:RecordDocketingMessage, a hyperlink is provided, but produces an error when used:



However, from this errant hyperlink, I can deduce that the schema for docket:RecordDocketingMessage is located in the schema folder as docket.xsd.

Maintaining relative links is troublesome in Word. Before we publish the final specification, we will make sure all the links work properly.

1. **RecordDocketingMessage – Missing Filing Lead/Connected Documents**

The elements filing:FilingLeadDocument and filing:FilingConnectedDocument are not provided on docket:RecordDocketingMessage. These are needed.

Perhaps you were thinking that these elements would be accessed through docket:CorrectedFiling ( the docket.xml example suggests that this may be the case). However, you had agreed to make docket:CorrectedFiling optional (i.e. “Modified the multiplicity to 0,1”) which means these document elements would not be available when docket:CorrectedFiling is not usxed.

Also, even though you said that you had modified the multiplicity to 0,1, in work draft 08, docket:CorrectedFiling is still minOccurs = 1.

The ECF 5 ReviewDocketingMessage needs to include two versions of documents and case information – one pre-review and one post-review. The post-review documents are included in docket:ReviewedLeadDocument and docket:ReviewedConnectedDocument. I renamed docket:CorrectedFiling to filing:FilingMessage to provide the pre-review documents and case information. I fixed the cardinality and it is now 0,1.

1. **CorrectedFiling Description**

Since the CoreFilingMessage is no more in ECF5, the description for CorrectedFiling should be revised; currently: “The complete CoreFilingMessage submitted for review with any corrections applied by the clerk prior to accepting into the court record”. (underline added).

I fixed the definition.

1. **Filing Party and Attorney Identifiers**

The RecordDocketingMessage example (i.e. docket.xml) shows a FilingPartyID value of 10:

<docket:RecordDocketingMessage … >

 <nc:DocumentIdentification>

 <nc:IdentificationID>10</nc:IdentificationID>

 </nc:DocumentIdentification>

 <ecf:FilingPartyID>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:FilingPartyID>

But who is filing party number 10?

The FilingAttorneyID is also 10 in Filing Lead and Connected Document.

The j:CaseInitiatingParty (i.e. John W. Doe, Defendant) has a FilingPartyID of 10. But so does the j:CaseRespondentParty (i.e. Jane Q Doe, Defendant).

So who is filing party ID 10? Is it John W. Doe or is it Jane Q. Doe?

I changed the IDs in this example so that they do not conflict.

1. **CaseInitiatingParty = Defendant ????**

In the RecordDocketingMessage example (i.e. docket.xml) the j:CaseInitiatingParty has CaseParticipantRoleCode “Defendant”. This is in contradiction to the definition of CaseInitiatingParty (see #2 ‘Type of Party’, above).

I changed the role code to “Plaintiff”

1. **Docket.xml Example**

This item addresses understanding of CorrectedFiling.

The RecordDocketingMessage example (i.e. docket.xml) contains following XML fragment, prior to the <docket:CorrectedFiling> element:

 <nc:DocumentIdentification>

 <nc:IdentificationID>10</nc:IdentificationID>

 </nc:DocumentIdentification>

 <ecf:FilingPartyID>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:FilingPartyID>

 <ecf:SendingMDELocationID>

 <nc:IdentificationID>http://example.com/court</nc:IdentificationID>

 </ecf:SendingMDELocationID>

 <ecf:ServiceInteractionProfileCode>urn:oasis:names:tc:legalxml-courtfiling:schema:xsd:WebServicesMessaging-5.0</ecf:ServiceInteractionProfileCode>

 <j:CaseCourt>

 <nc:OrganizationIdentification>

 <nc:IdentificationID>10</nc:IdentificationID>

 </nc:OrganizationIdentification>

 <nc:OrganizationUnitName>Municipal Court</nc:OrganizationUnitName>

 </j:CaseCourt>

 <nc:DocumentPostDate>

 <nc:DateTime>2008-07-07T13:47:42.0Z</nc:DateTime>

</nc:DocumentPostDate>

The first few elements within docket:CorrectedFiling are:

 <nc:DocumentIdentification>

 <nc:IdentificationID>123456ABC</nc:IdentificationID>

 </nc:DocumentIdentification>

 <ecf:ElectronicServiceInformation>

 <ecf:ReceivingMDELocationID>

 <nc:IdentificationID>http://example.com/efsp2</nc:IdentificationID>

 </ecf:ReceivingMDELocationID>

 <ecf:ReceivingMDEProfileCode>urn:oasis:names:tc:legalxml-courtfiling:schema:xsd:WebServicesMessaging-5.0</ecf:ReceivingMDEProfileCode>

 <ecf:ServiceRecipientID>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:ServiceRecipientID>

 </ecf:ElectronicServiceInformation>

 <ecf:FilingPartyID>

 <nc:IdentificationID>10</nc:IdentificationID>

 </ecf:FilingPartyID>

 <ecf:SendingMDELocationID>

 <nc:IdentificationID>http://example.com/efsp1</nc:IdentificationID>

 </ecf:SendingMDELocationID>

 <ecf:ServiceInteractionProfileCode>urn:oasis:names:tc:legalxml-courtfiling:schema:xsd:WebServicesMessaging-5.0</ecf:ServiceInteractionProfileCode>

 <j:CaseCourt>

 <nc:OrganizationIdentification>

 <nc:IdentificationID>10</nc:IdentificationID>

 </nc:OrganizationIdentification>

 <nc:OrganizationUnitName>Municipal Court</nc:OrganizationUnitName>

 </j:CaseCourt>

 <nc:DocumentInformationCutOffDate>

 <nc:DateTime>2008-07-07T13:47:42.0Z</nc:DateTime>

 </nc:DocumentInformationCutOffDate>

 <nc:DocumentPostDate>

 <nc:DateTime>2008-07-07T13:47:42.0Z</nc:DateTime>

 </nc:DocumentPostDate>

Differences that appear to be corrections within CorrectedFiling are highlighted in yellow.

So how should these differences (i.e. corrections) be interpreted?

Starting with CorrectedFiling’s:

 <nc:DocumentIdentification>

 <nc:IdentificationID>123456ABC</nc:IdentificationID>

 </nc:DocumentIdentification>

Is this to be understood as the nc:DocumentIdentification/nc:Identification provided in the filing:FilingMessage was incorrect, and was adjusted in clerk review to be the value ‘123456ABC’?

If so, what was the original filing:FilingMessage nc:DocumentIdentification/nc:Identification value? Is it ‘10’?

CorrectedFiling added the ecf:ElectronicServiceInformation structure. Is this understood as the original filing:FilingMessage did not request any service, but the clerk added service for Service Recipient 10? Who is ServiceRecipientID 10? ServiceRecipientID is not addressed in section 6.2 Identifier Rules.

The SendingMDELocationID provided on the filing:FilingMessage (i.e. ‘http://example.com/court’) was revised/corrected in clerk review to be ‘http://example.com/efsp1 ‘.

Finally for the example fragment, the original filing:FilingMessage did not provide any nc:DocumentInformationCutOffDate, but this was added in clerk review.

This is just a quickly manufactured example which, for the reasons you have identified, may not be completely consistent. Please feel free to provide a better example if you like.