S-100 Maintenance Proposals

Part 10c (HDF5)
Part 8 (Gridded data)

S100WG4 / S102PT
25 February – 1 March 2019

Raphael Malyankar

Eivind Mong

Sponsored by NOAA

Overview

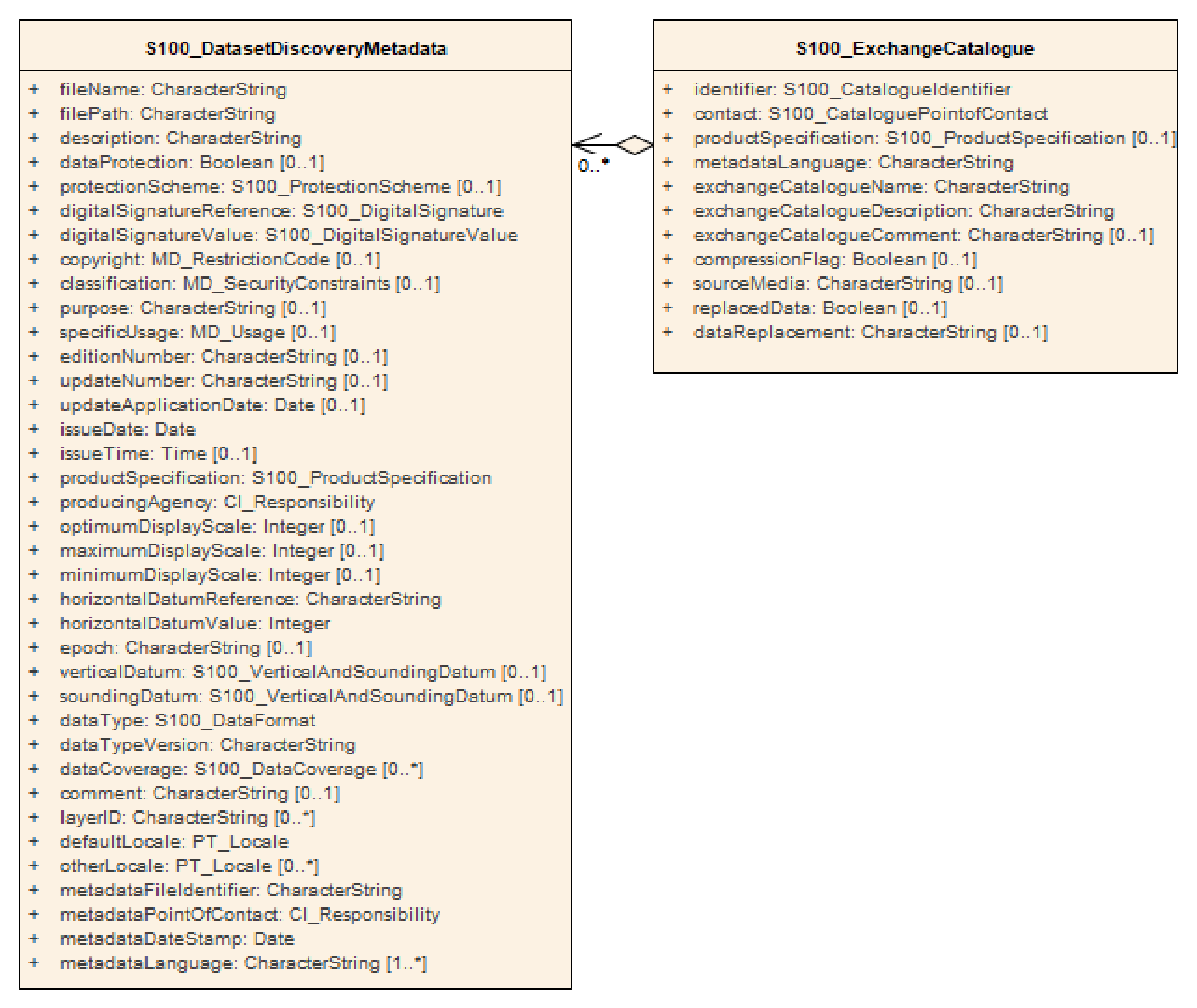
- Proposal 1:
  - Provisions for use of HDF5 “File Families.”
- Proposal 2:
  - Provisions for specifying the “data sample point” location in the cell.
  - Miscellaneous clarifications in Parts 10c (HDF5) and 8 (Imagery and Gridded Data).

S100WG4-4.12 HDF5 File Families

- An HDF5 file family is one *logical* file mapped to more than one *physical* files.
- Use case:
  - For some types of data, the amount of data can be several Gb or even Tb.
  - With file families, an HO could in theory build their datasets as big as they want and still meet a requirement imposing a physical file size limit.
- This proposal describes the S-100 metadata and related implementation for Product Specifications which allow file families.
- Product Specifications may have to be written to accommodate large datasets.
  - Determinations of and limits on maximum size are out of scope for the present proposal. OEMs may desire a lower limit (10 MB or 256MB) depending on method of transmission.
  - The present proposal could probably be adapted to apply to (separate) tiles or otherwise partitioned datasets.

Considerations

- Validation of the exchange set requires knowing what physical files are supposed to be in the exchange set.
  - The S-100 metadata model does not include a file count attribute. There is supposed to be a different discovery metadata block for each file (dataset or support). Generally, that suffices as an implicit count.
  - A different discovery metadata block for each physical file in an HDF5 file family would be duplicative except for physical file name and digital signature.
  - In principle there can be more than one dataset in an exchange set – i.e., multiple sets of file families. So the number of files in a “file family” cannot be placed in exchange set metadata – it has to be in dataset discovery metadata.
- This proposal describes the metadata for a file family.
  - Product specifications are expected to add this metadata as an extension to the standard S-100 metadata described in Part 4a, if they allow file families.
  - Product specifications must extend S-100 generic schemas to add it. (See S-97.)
- There is also some implementation guidance for developers added to Part 10c.



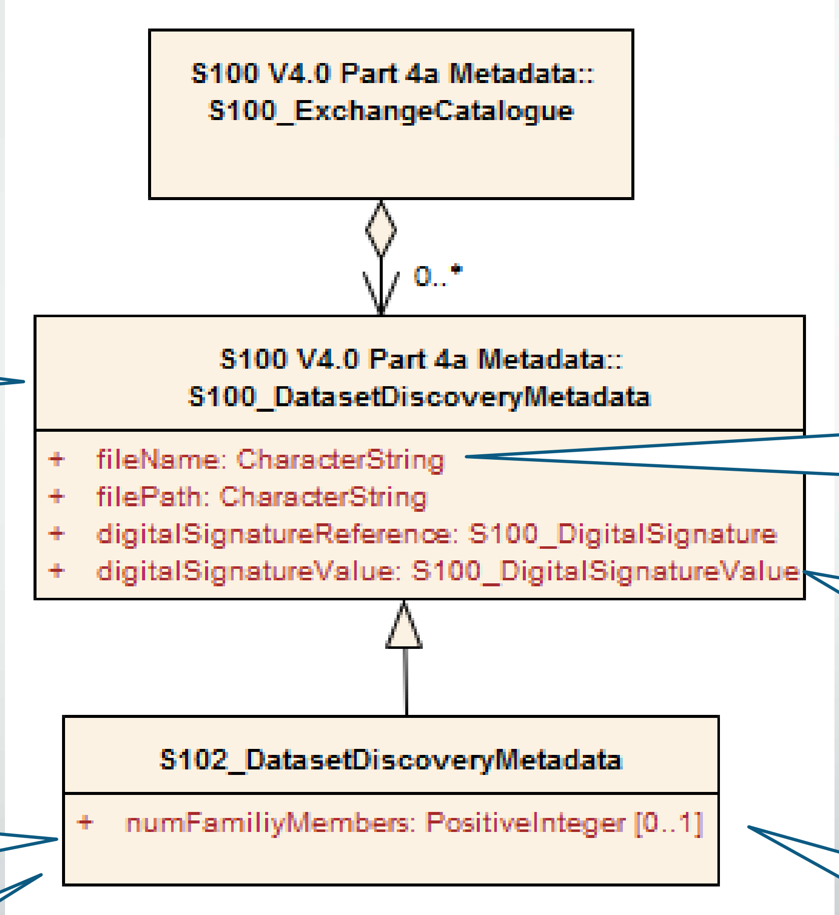
S100_DatasetDiscoveryMetadata

- + fileName: CharacterString
- + filePath: CharacterString
- + description: CharacterString
- + dataProtection: Boolean [0..1]
- + protectionScheme: S100_ProtectionScheme [0..1]
- + digitalSignatureReference: S100_DigitalSignature
- + digitalSignatureValue: S100_DigitalSignatureValue
- + copyright: MD_RestrictionCode [0..1]
- + classification: MD_SecurityConstraints [0..1]
- + purpose: CharacterString [0..1]
- + specificUsage: MD_Usage [0..1]
- + editionNumber: CharacterString [0..1]
- + updateNumber: CharacterString [0..1]
- + updateApplicationDate: Date [0..1]
- + issueDate: Date
- + issueTime: Time [0..1]
- + productSpecification: S100_ProductSpecification
- + producingAgency: CI_Responsibility
- + optimumDisplayScale: Integer [0..1]
- + maximumDisplayScale: Integer [0..1]
- + minimumDisplayScale: Integer [0..1]
- + horizontalDatumReference: CharacterString
- + horizontalDatumValue: Integer
- + epoch: CharacterString [0..1]
- + verticalDatum: S100_VerticalAndSoundingDatum [0..1]
- + soundingDatum: S100_VerticalAndSoundingDatum [0..1]
- + dataType: S100_DataFormat
- + dataTypeVersion: CharacterString
- + dataCoverage: S100_DataCoverage [0..*]
- + comment: CharacterString [0..1]
- + layerID: CharacterString [0..*]
- + defaultLocale: PT_Locale
- + otherLocale: PT_Locale [0..*]
- + metadataFileIdentifier: CharacterString
- + metadataPointOfContact: CI_Responsibility
- + metadataDateStamp: Date
- + metadataLanguage: CharacterString [1..*]

S100_ExchangeCatalogue

- + identifier: S100_CatalogueIdentifier
- + contact: S100_CataloguePointofContact
- + productSpecification: S100_ProductSpecification [0..1]
- + metadataLanguage: CharacterString
- + exchangeCatalogueName: CharacterString
- + exchangeCatalogueDescription: CharacterString
- + exchangeCatalogueComment: CharacterString [0..1]
- + compressionFlag: Boolean [0..1]
- + sourceMedia: CharacterString [0..1]
- + replacedData: Boolean [0..1]
- + dataReplacement: CharacterString [0..1]

Proposal in a nutshell



S100 V4.0 Part 4a Metadata::
S100_ExchangeCatalogue

0..*

S100 V4.0 Part 4a Metadata::
S100_DatasetDiscoveryMetadata

+ fileName: CharacterString
+ filePath: CharacterString
+ digitalSignatureReference: S100_DigitalSignature
+ digitalSignatureValue: S100_DigitalSignatureValue

S102_DatasetDiscoveryMetadata

+ numFamiliyMembers: PositiveInteger [0..1]

There is a single dataset discovery metadata instance for each **logical** dataset file.

The filename attribute names the **logical** dataset file.
(myfile.hdf5, not myfile_0.hdf5)

The digital signature is computed using all the physical files in the file family, in order.

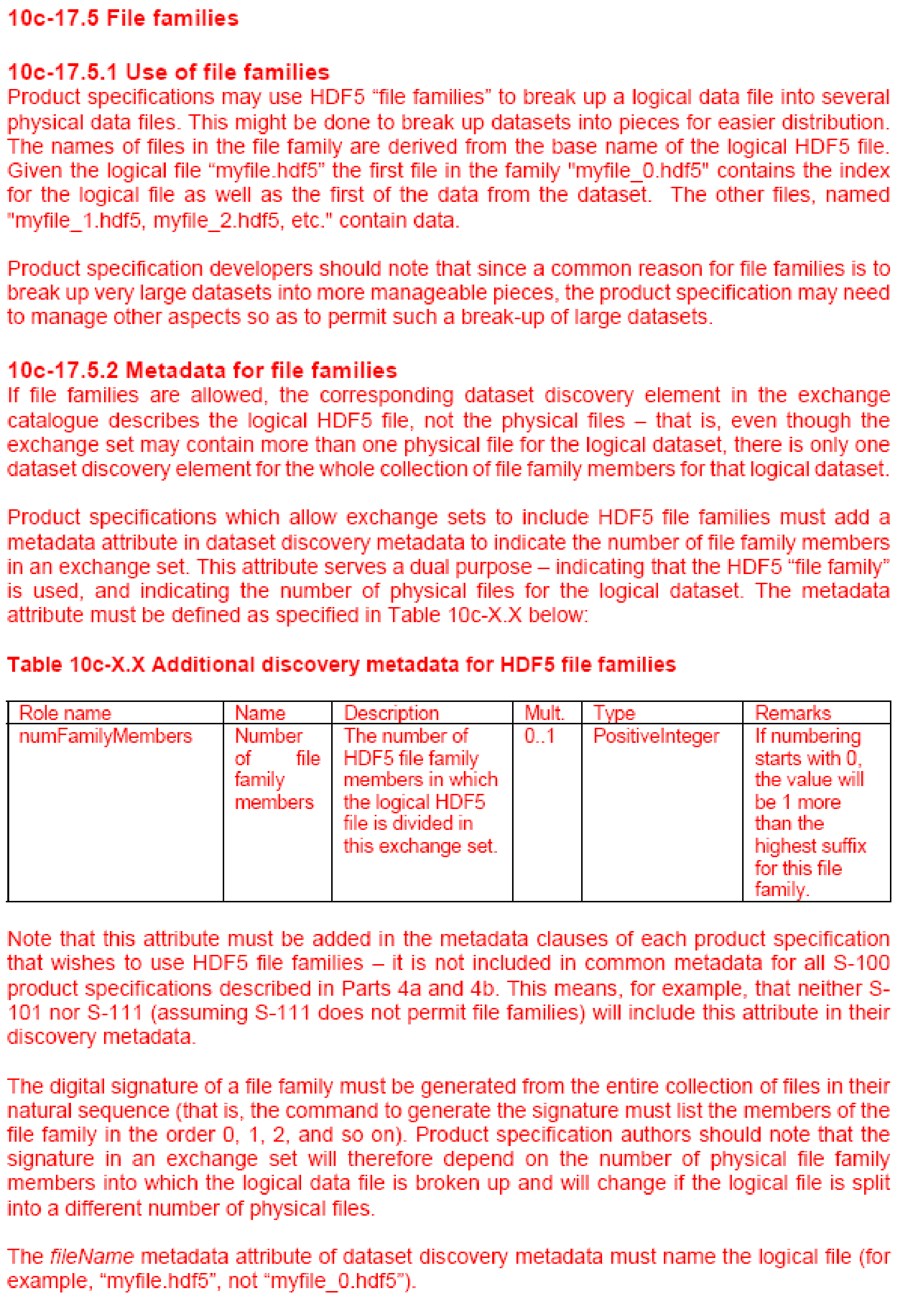
Product Spec. extends S-100 dataset discovery metadata with attribute *numFamilyMembers* (containing the number of **physical** files for the logical dataset)

It will be used by a small minority of product specifications, so it is not added to common S-100 discovery metadata in Part 4a.

If this attribute is not present, file families are not being used.

Extract from exchange catalogue model in product specification showing relevant classes and attributes

Details – file families



10c-17.5 File families

10c-17.5.1 Use of file families

Product specifications may use HDF5 “file families” to break up a logical data file into several physical data files. This might be done to break up datasets into pieces for easier distribution. The names of files in the file family are derived from the base name of the logical HDF5 file. Given the logical file “myfile.hdf5” the first file in the family "myfile_0.hdf5" contains the index for the logical file as well as the first of the data from the dataset. The other files, named "myfile_1.hdf5, myfile_2.hdf5, etc." contain data.

Product specification developers should note that since a common reason for file families is to break up very large datasets into more manageable pieces, the product specification may need to manage other aspects so as to permit such a break-up of large datasets.

10c-17.5.2 Metadata for file families

If file families are allowed, the corresponding dataset discovery element in the exchange catalogue describes the logical HDF5 file, not the physical files – that is, even though the exchange set may contain more than one physical file for the logical dataset, there is only one dataset discovery element for the whole collection of file family members for that logical dataset.

Product specifications which allow exchange sets to include HDF5 file families must add a metadata attribute in dataset discovery metadata to indicate the number of file family members in an exchange set. This attribute serves a dual purpose – indicating that the HDF5 “file family” is used, and indicating the number of physical files for the logical dataset. The metadata attribute must be defined as specified in Table 10c-X.X below:

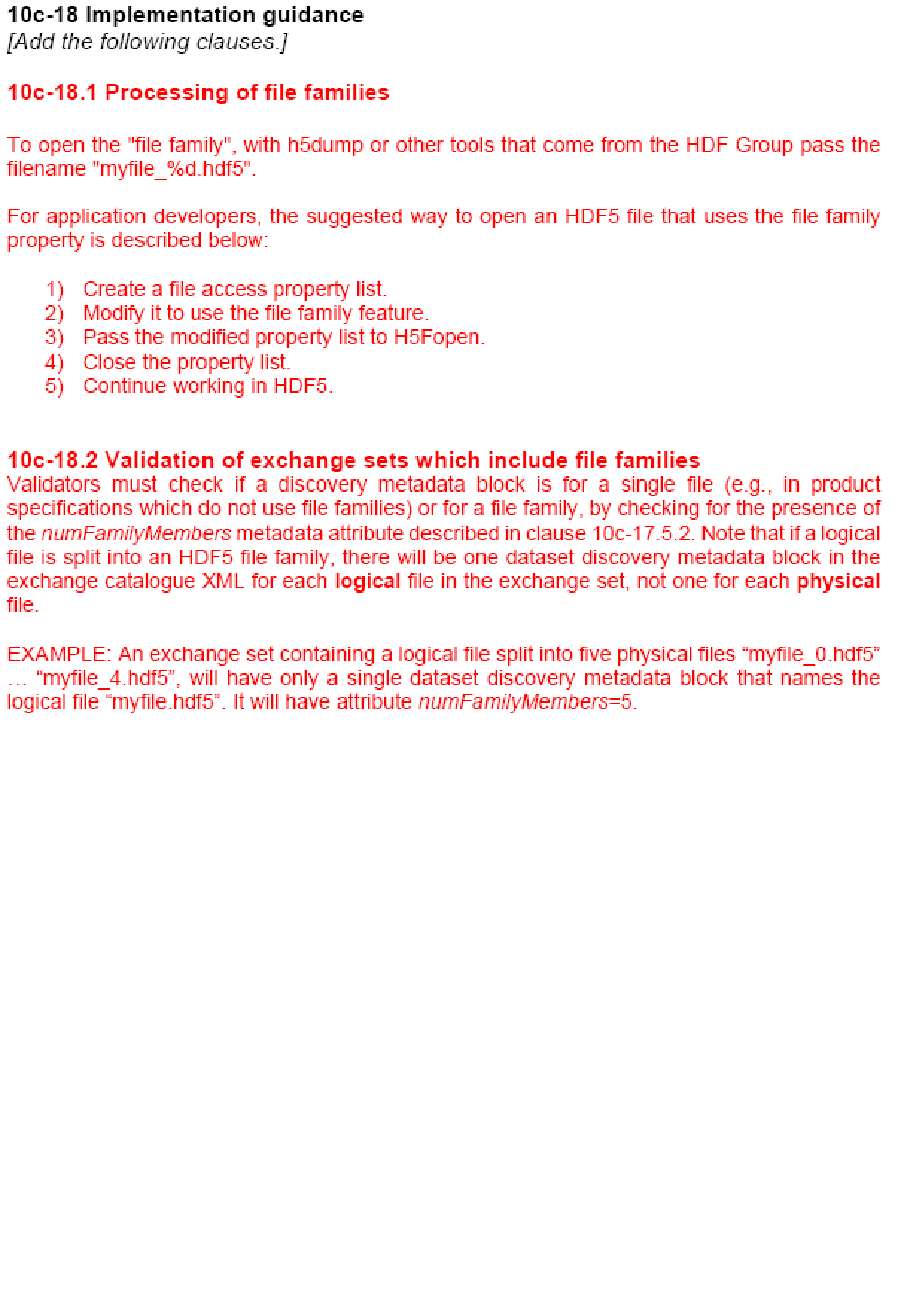
Table 10c-X.X Additional discovery metadata for HDF5 file families

Role name	Name	Description	Mult.	Type	Remarks
numFamilyMembers	Number of file family members	The number of HDF5 file family members in which the logical HDF5 file is divided in this exchange set.	0..1	PositiveInteger	If numbering starts with 0, the value will be 1 more than the highest suffix for this file family.

Note that this attribute must be added in the metadata clauses of each product specification that wishes to use HDF5 file families – it is not included in common metadata for all S-100 product specifications described in Parts 4a and 4b. This means, for example, that neither S-101 nor S-111 (assuming S-111 does not permit file families) will include this attribute in their discovery metadata.

The digital signature of a file family must be generated from the entire collection of files in their natural sequence (that is, the command to generate the signature must list the members of the file family in the order 0, 1, 2, and so on). Product specification authors should note that the signature in an exchange set will therefore depend on the number of physical file family members into which the logical data file is broken up and will change if the logical file is split into a different number of physical files.

The *fileName* metadata attribute of dataset discovery metadata must name the logical file (for example, “myfile.hdf5”, not “myfile_0.hdf5”).



10c-18 Implementation guidance

[Add the following clauses.]

10c-18.1 Processing of file families

To open the "file family", with h5dump or other tools that come from the HDF Group pass the filename "myfile_%d.hdf5".

For application developers, the suggested way to open an HDF5 file that uses the file family property is described below:

1) Create a file access property list.
2) Modify it to use the file family feature.
3) Pass the modified property list to H5Fopen.
4) Close the property list.
5) Continue working in HDF5.

10c-18.2 Validation of exchange sets which include file families

Validators must check if a discovery metadata block is for a single file (e.g., in product specifications which do not use file families) or for a file family, by checking for the presence of the *numFamilyMembers* metadata attribute described in clause 10c-17.5.2. Note that if a logical file is split into an HDF5 file family, there will be one dataset discovery metadata block in the exchange catalogue XML for each **logical** file in the exchange set, not one for each **physical** file.

EXAMPLE: An exchange set containing a logical file split into five physical files “myfile_0.hdf5” … “myfile_4.hdf5”, will have only a single dataset discovery metadata block that names the logical file “myfile.hdf5”. It will have attribute *numFamilyMembers*=5.

Conclusion – HDF5 File families

- Comments and questions?