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).

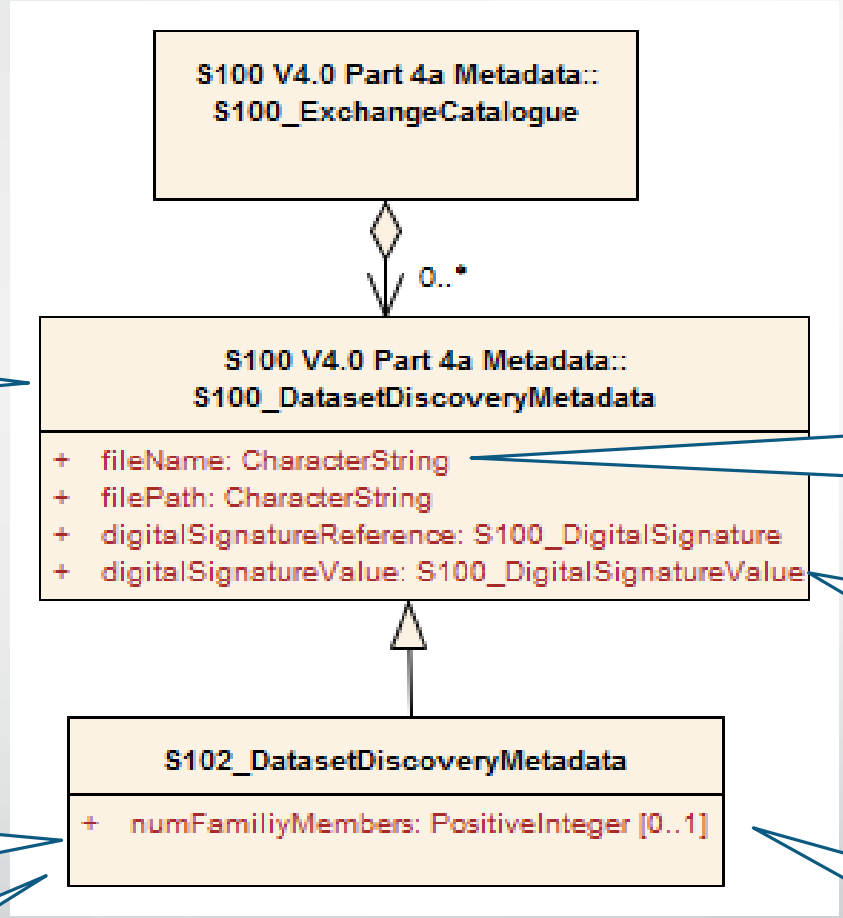
S100WG4-4.12 HDF5 File Families

- 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.

Proposal in a nutshell



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.

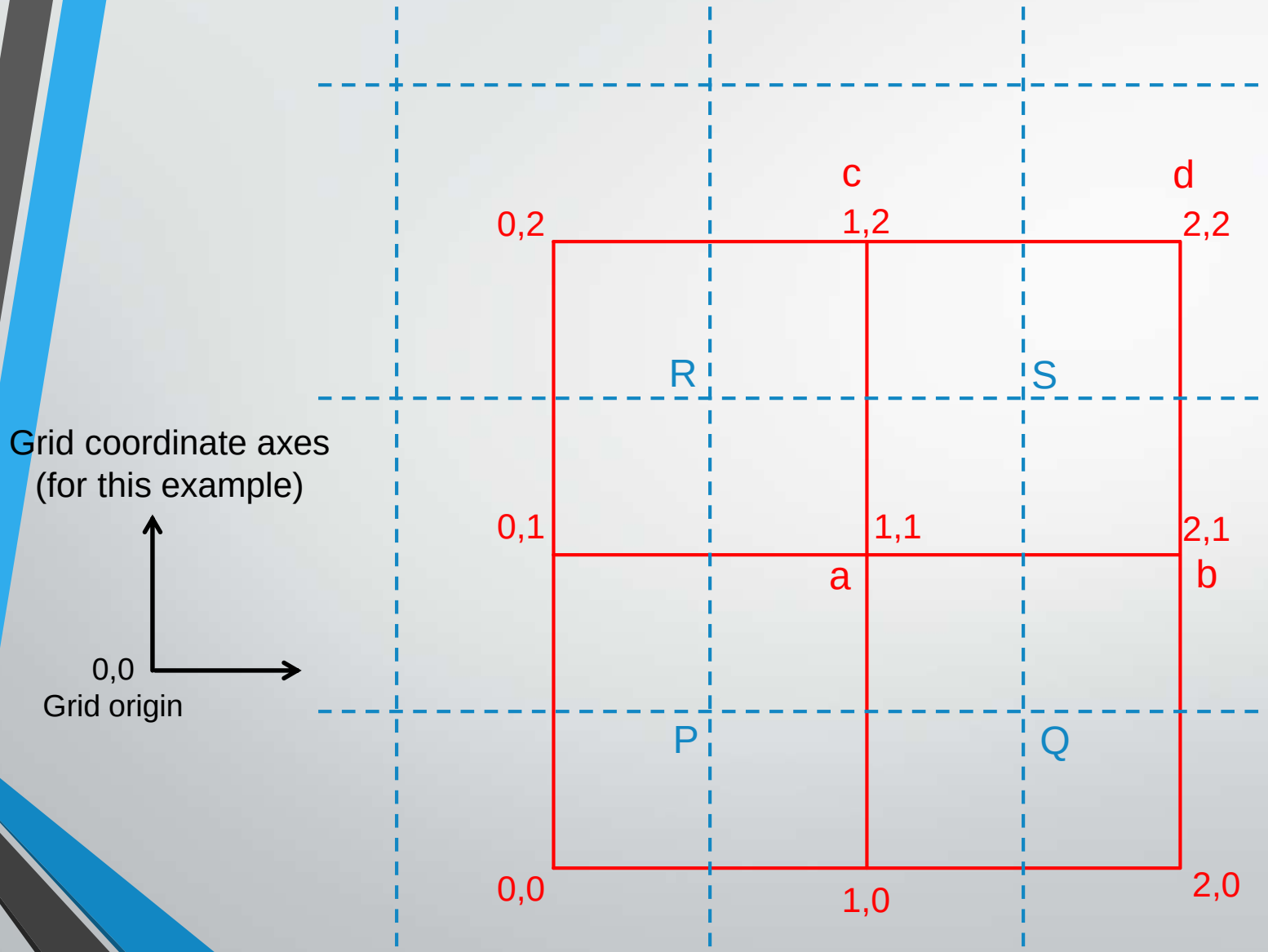
Conclusion – HDF5 File families

- Comments and questions?

S100WG4-4.14 Location of data point in cell

- ISO 19123 and S-100 make the locations of the data points in grids coincident with the vertices of the grid (“grid points”).
- Project teams want the ability to specify where the data (sample) point is located in the grid cells. It matters for interpolation and portrayal.
  - Encoding gridded data with the data point at a corner or the center of a grid cell are both common practices.
  - Allows non-overlapping data points for adjacent grid features (e.g., for adjacent ENC cells).
- This proposal defines two (mutually exclusive) attributes to indicate this.
  - The simple approach: dataOffsetCode – an enumeration attribute to indicate whether the lower/upper left/right corner, or the cell center, is the data point. This is intended for the most common cases – cell corner/center in 2-D grids.
  - The complete approach: dataOffsetVector – the relative position of the data point in a cell (relative to the cell size in each dimension). This works for all dimensions. It also allows the data point to be more precisely positioned if needed.
- Product specifications are expected to pick one or the other depending on their needs. If a product specification does not use either, the default ISO 19123/S-100 location (LL corner) applies.

Attribute - dataOffsetCode



Default situation

a,b,c,d: grid cell

a: grid point and also point location of data for cell

P,Q,R,S: (implicitly) sample space of grid point a

Proposal - dataOffsetCode

dataOffsetCode=1 (XMin, YMin) [“LL”]

a: data point location

dataOffsetCode=2 (XMax, YMax) [“UR”]

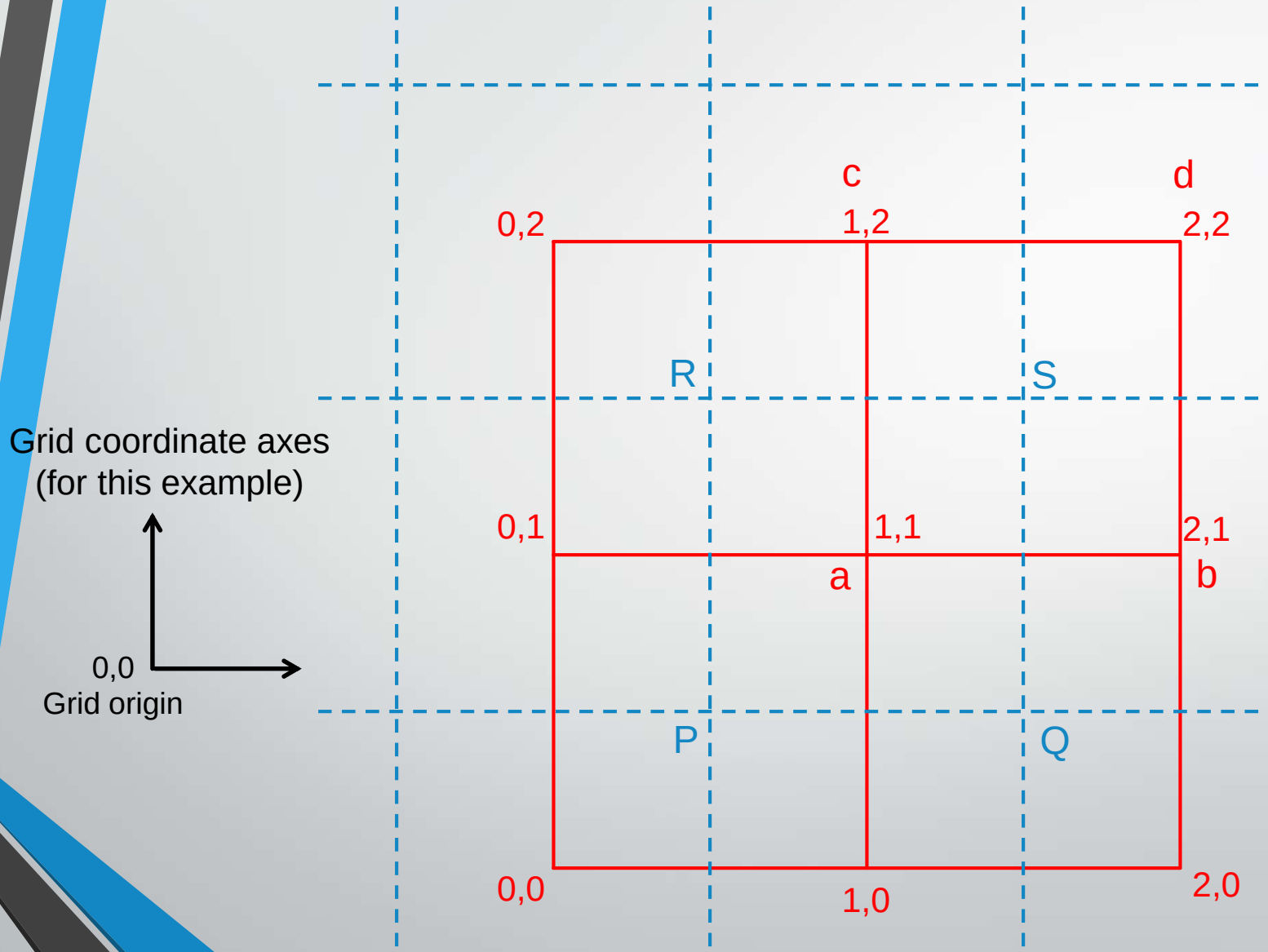
d: data point location

dataOffsetCode=5 (bary-center) [grid cell center]

S: data point location

3,4: other corners

Attribute - dataOffsetVector



Default situation

a,b,c,d: grid cell

a: grid point and also point location of data for cell

P,Q,R,S: (implicitly) sample space of grid point a

Proposal - dataOffsetVector

dataOffsetVector=(0.0, 0.0)

a: data point location

dataOffsetVector=(1.0, 1.0)

d: data point location

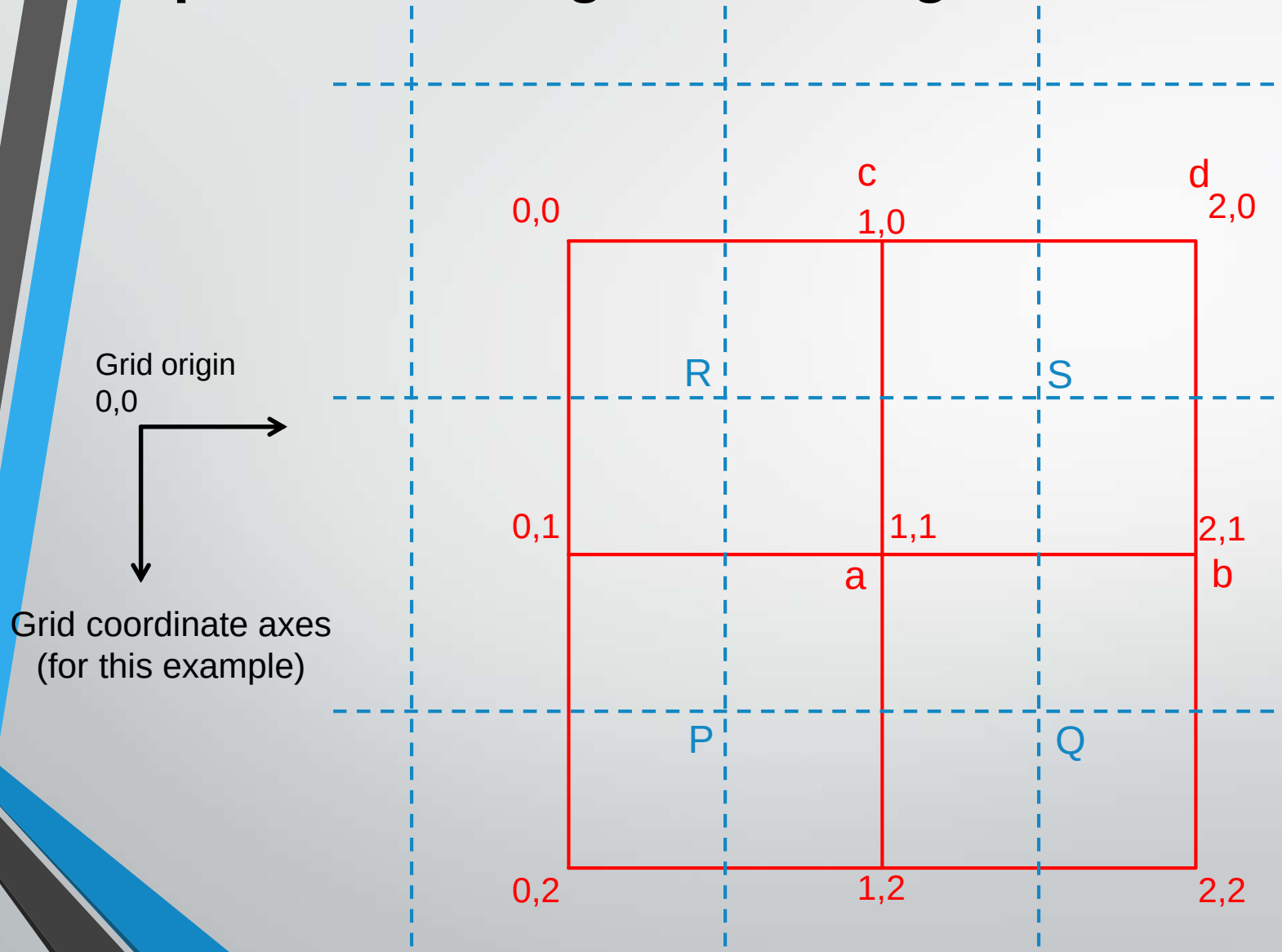
dataOffsetVector=(0.5,0.5)

S: data point location

Any position in the range [0.0, 0.0] to [1.0, 1.0]

Works for N dimensions

Dependence on grid axis origin/direction



Default situation

a,b,c,d: grid cell

c: grid point and also point location of data for cell abcd

Proposal - dataOffsetCode

dataOffsetCode=1 (XMin, YMin) [“LL”]

c: data point location

dataOffsetCode=2 (XMax, YMax) [“UR”]

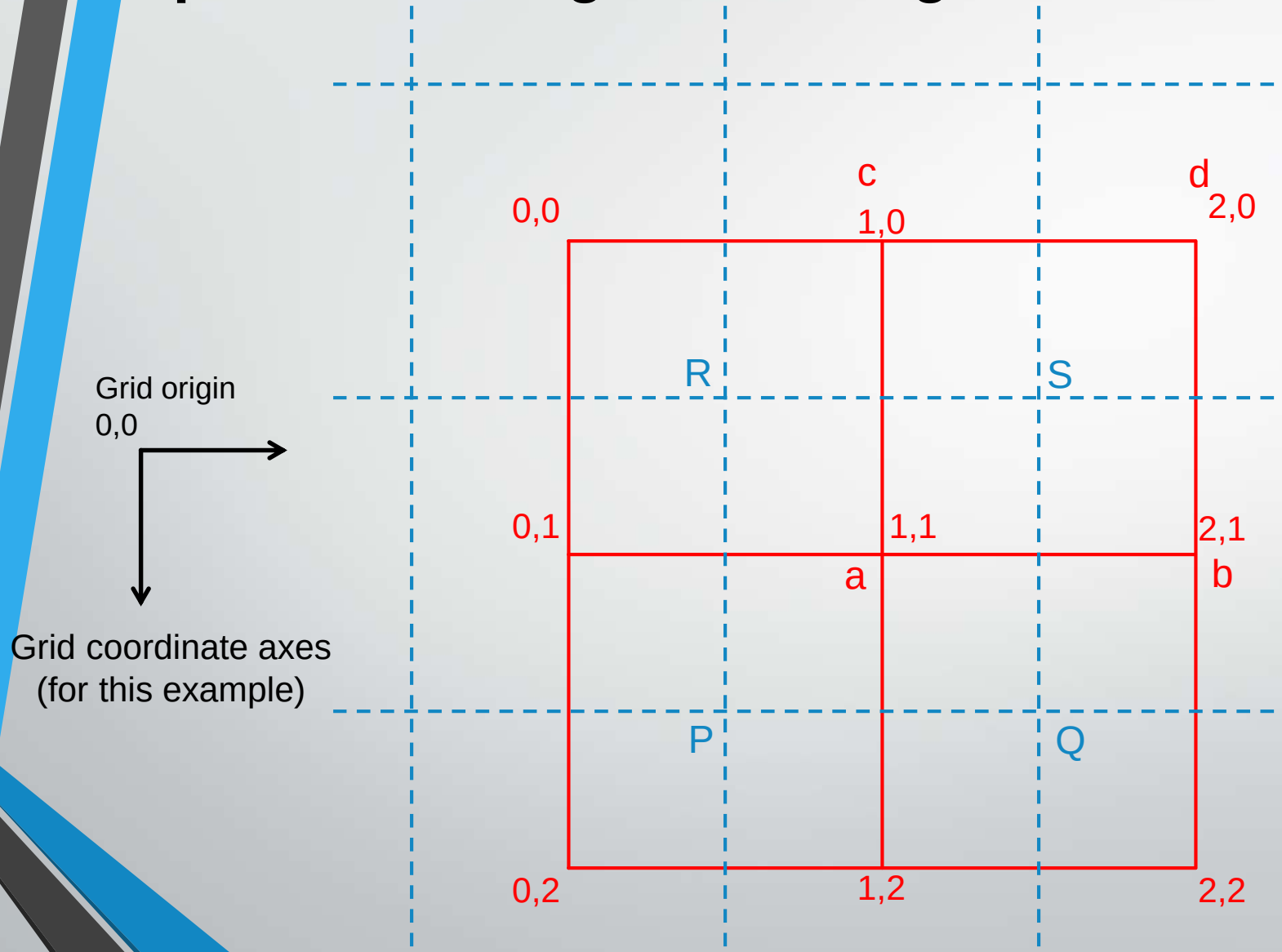
b: data point location

dataOffsetCode=5 (bary-center) [grid cell center]

S: data point location

3,4: other corners

Dependence on grid axis origin/direction



Default situation

a,b,c,d: grid cell

c: grid point and also point location of data for cell abcd

Proposal - dataOffsetVector

dataOffsetVector=(0.0, 0.0)

c: data point location

dataOffsetVector=(1.0, 1.0)

b: data point location

dataOffsetVector=(0.5,0.5)

S: data point location

Any position in the range [0.0, 0.0] to [1.0, 1.0]

Works for N dimensions

Things to consider

- Are all 4 (2-D) corner codes necessary in practice, or just LL, UR, and center?
  - If only the two extrema and the center – can use codes for 3-D grids as well.
- The underlying ISO standard describes the bilinear, biquadratic, and bicubic interpolation in terms of combinations of the values at the *vertices* of grid cells, implying that the data values are supposed to be located at the corners of the grid cells, i.e., exactly at the grid points.
- It is possible to achieve equivalent results by defining appropriate interpolation parameters instead.
  - The project teams prefer using an attribute, as being simpler than interpolation.

Details

10c-9.6 Feature container group

[Add the following attribute to Table 10c-10 in each of the sections for dataCodingFormat = 2 (Regularly-gridded arrays), 3 (Ungeorectified gridded arrays) , 5 (Irregular grid), 6 (Variable cell size). Add 10c-9.6.1 to explain the use of the new attributes.]

Name	Camel case	Mult.	Data Type	Remarks and/or units
Offset of data point in cell	dataOffsetCode	0..1	Enumeration	1: XMin, YMin ("Lower left") corner ("Cell origin") 2: XMax, YMax ("Upper right") corner 3: XMax, YMin ("Lower right") corner 4: XMin, YMax ("Upper left") corner 5: Barycenter (centroid) of cell
Offset of data point in cell as vector	dataOffsetVector	0..1	Float	Array (1-D) 0..D-1 where D is the value of the dimension attribute Values must be real numbers in the range [0,1].

10c-9.6.1 Location of data point within cell

Product specifications may require their data products to indicate the relative location of the data point corresponding to a grid cell in relation to the corners of the cell. The location can be indicated using either the dataOffsetCode or dataOffsetVector attribute. These attributes can be used only with grid-based coverages and not with time series, TIN, or moving platform data. Product specifications may use either dataOffsetCode or dataOffsetVector but not both.

Product specifications in which the data point is located at the (XMin, YMin) grid point need not use either dataOffsetCode or dataOffsetVector.

The attribute dataOffsetCode can be used only with two-dimensional grids. It indicates whether the data point is one of the four cell corners or the centre of the cell. Note that the definitions of the codes indicting the corners are in terms of X and Y grid coordinates relative to the grid origin. (This means that in a grid with its X axis directed from east to west and Y axis from north to south the "lower left" corner is different from the "lower left" corner in a grid with X axis directed west to east and Y axis south to north.)

The attribute dataOffsetVector is intended for use with higher-dimension grids or in cases where the data point location is not at one of the corners or the centre of the cell. The values in this array indicate the relative offset along each axis of the data point from the grid point whose grid coordinates are closest to those of the grid origin. In a two-dimensional grid, this will be the point with smallest X and Y grid coordinates. Again, it should be noted that the direction of the axes and the location of the grid origin determines which corner is the cell origin. Each offset is relative to the dimension of the cell along the corresponding axis. The order of values in dataOffsetVector must correspond to the order of axes in the axisNames array (Table 10c-9).

Conclusion – data point location in grid cell

- Comments and questions?

Details – data coding format

10c-10.4 Data coding format

Item	Name	Description	Code	Remarks
Enumeration	S100_HDF_ DataCodingFormat	Data coding formats for S-100 HDF5 data		
Literal	fixedStations	Data at multiple discrete fixed point locations.	1	
Literal	regularGrid	Data at grid points forming a regular grid with constant cell spacing.	2	Regular grids are commonly composed of perpendicularly crossing lines of equal spacing on each dimension, creating square or rectangular cells.
Literal	ungeorectifiedGrid	Data that does not include any information that can be used to determine a cell's geographic coordinate values, or in which cell spacing is variable, and there is no predefined association between one cell's location and that of another.	3	For example, a digital perspective aerial photograph without georectification information included
Literal	movingPlatform	Data at sequential discrete point locations of a moving sensor platform.	4	
Literal	irregularGrid	Data distributed over a grid with uniform cell spacing but irregular overall shape.	5	The irregularity of shape may consist of non-rectangular coverage area or relatively large regions which are not populated with data.
Literal	variableCellSize	Variable-density grid containing one or more regions with cell spacing that is a whole multiple of a common minimum uniform cell spacing.	6	The shape of the overall grid may be non-rectangular.
Literal	TIN	Triangulated irregular network.	7	A TIN is a representation of a continuous surface consisting entirely of triangular facets. The vertices at the corners of each triangle are shared with the adjacent triangle. These vertices form the control points of the coverage function.

Details – grid cell structure clarification

Part 8 Imagery and Gridded Data

8-6.2 Point Sets, Grids, and TINs

…

8-6.2.8 Grid cell structure

S-100 utilizes the same view of grid cell structure as Section 8.2.2 of ISO 19123. The grid data in S-100 grid coverages are nominally situated exactly at the grid points defined by the grid coordinates. The grid points are therefore the "sample points." Data values at a sample point represent measurements over a neighbourhood of the sample point. This neighbourhood is assumed to extend a half-cell in each dimension. The effect is that the sample space corresponding to each grid point is a cell centred at the grid point.

Note that applying interpolation methods to a coverage means that the value of a data characteristic at a location between grid points may be different from that at any or all of the grid points which are its nearest neighbours.

Some data products may find it convenient to use nominal locations of data measurements that do not coincide with grid points as outlined above. Part 10c provides a method for encoding such data products by selecting one of the corners of the cell or by defining a standard offset to be applied to the default grid point locations in order to determine the nominal locations of the data values.