Spatio-Temporal Big Data Standards: Status and Progress

Peter Baumann
Computer Science & Electrical Engineering
Constructor University
Bremen, Germany
email: pbaumann@constructor.university

***Abstract*–In standardization, the term *coverage* captures the digital representation of space/time-varying phenomena. Coverages are supported by a mature set of standards, maintained in a continuous cooperation of the International Organization for Standardization (ISO) and Open Geospatial Consortium (OGC), with manifold uptake and implementation. At its heart is the OGC/ISO Coverage Implementation Schema (CIS) data standard. We give a condensed overview of the CIS standard and its current progress, looking at the ISO 19123-1 concepts and their realization with ISO 19123-2. We do this in our capacity as primary editor of the standards discussed.**

Keywords- coverages, datacubes, standards, ISO, OGC, rasdaman.

I. INTRODUCTION

Phenomena observed on, in, or above Earth often represent *fields* as defined in physics (e.g., quantum field theory [8]): some quantity that has a value for each point in space and time within some region. In other words: the quantity varies in space and time. Examples include the Earth's magnetic field, surface wind maps, and river water temperature at some location; Figure 1 shows a kaleidoscope of data from various geo application domains.

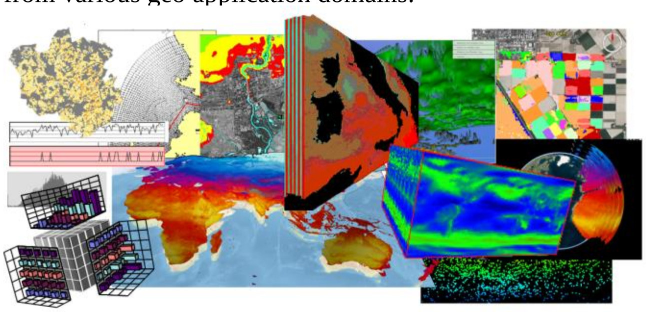


Figure 1. Basic building blocks of a coverage.

Such fields are multi-dimensional by nature – in the above examples we find 4-D (four-dimensional) x/y/z/t for the magnetic field, 3-D x/y/t for the wind map, and 1-D for the water temperature timeseries. Obviously, the dimension axes can be spatial or temporal; however, they even can have further dimensions, such as a spectral dimension for wave frequencies occurring; a second time axis, as used in weather forecasting; a species axis for measuring habitat changes in a region over time.

Mathematically, such a field can be seen as a function which assigns a value (from its range) to every point in the region where the function is defined (its domain). In standardization, the term *coverage* subsumes digital representations of such space/time varying phenomena. Technically, coverages encompass regular and irregular grids, point clouds, and general meshes. Most notably, they serve to represent raster data and spatio-temporal datacubes. To cite common phrases, such data typically constitute "Big Data", which are "too big to transport", so that processing requires to "ship code to the data".

The central standard is the Open Geospatial Consortium (OGC) *Coverage Implementation Schema* (CIS) [27] and the parallel International Organization for Standardization (ISO) 19123-2 [11], likewise nicknamed CIS. They are embedded in a larger ecosystem of data and service standards. In this contribution, we only look at the coverage data standards. Table 1 shows the correspondence of ISO and OGC coverage standards; see also the overview in [33].

Recently, these standards have undergone a revision and now are better structured (cf. Table I):

- *conceptual level*: ISO 19123-1 / OGC Abstract Topic (AT) 6.1 defines the information concepts, together with the pertaining terminology;
- *logical level*: ISO 19123-2 Clauses 5 to 10 / OGC CIS defines concrete data structures as object classes;
- *physical level*: ISO 19123-2 Clause 11 and 12 / OGC CIS plus further separate encoding standards define the mapping of logical-level data to byte streams such as XML, JSON, GeoTIFF, NetCDF, JPEG2000, etc.

ISO 19123-1 [10], which defines coverage concepts and terms, was adopted in 2023 replacing outdated 19123:2005. Several reasons prompted this evolution: difficult to understand; errors and omissions, such as excluding 1-D; definitions not state of the art, such as rasters defined as "corresponding to the display on a cathode ray tube"; mixed conceptual, logical, and physical levels making comprehension difficult.

TABLE I. CORRESPONDENCE OF OGC AND ISO COVERAGE STANDARDS.

ISO	OGC	contents
19123-1 [10]	Abstract Topic 6.1 [28]	Coverage data model: concepts & terminology
19123-2 [11]	CIS [27]	Coverage Implementation Schema
19123-3 [12]	Abstract Topic 6.3 [29]	Coverage processing model: concepts & terminology, based on OGC WCPS [26]

Consequently, 19123:2005 got split and replaced by two parts: 19123-1 [10] establishes the conceptual model using interfaces describing the high-level observable behavior of a coverage object, leaving implementation details open. Such detail is provided by 19123-2 [11] which contains the logical model and – clearly separated – the physical-level encoding. The standard is organized into packages resembling self-contained units where each one establishes a particular coverage concept.

The author is active OGC contributor since 2004 and in this capacity main editor of the currently 23 coverage / datacube / Web Coverage Service (WCS) standards [26]-[29][31], OGC delegate to ISO, ISO project lead / editor of the 19123-1/2/3 family of coverage standards [10]-[12], and German delegate and WCS drafting team member for EU INSPIRE, the European legal framework for a common spatial data infrastructure. Further, he is initiator and co-editor of ISO SQL/MDA (Multi-Dimensional Arrays) [9][19].

The remainder of this paper is organized as follows. In Section II, we present the concepts and terminology of coverages, followed by the concrete, implementation-oriented coverage structures in Section III. A brief lookout on a data language tailored to coverage analytics is given in Section IV. Related coverage standards are discussed in Section V. Section VI provides a summary.

II. COVERAGE CONCEPTUAL MODEL

The notion of a field as a function C: $D \rightarrow V$ suggests a rather simple definition of a coverage, plus an access method: just evaluate the function at any position $p \in D$, yielding $C(p) = v \in V$. As per ISO 19107 [14], this is called the *evaluate function*, commonly denoted as

$evaluate_C: D \rightarrow V,\ evaluate_C(p) = v$

While this is conceptually elegant, it is normally highly inefficient to ask for single coordinates, so this is not the kind of functionality specifically supported in coverage services; rather, extraction and processing of larger regions is common, e.g., in WCS [31] and Web Coverage Processing Service (WCPS) [3][26].

Actually, the above function definition needs an extension to allow multiple values for a location:

$$evaluate_C: D \rightarrow P(V),\ evaluate_C(p) = \bigcup_{f \in C} f.contains(p)$$

where $P(V)$ denotes the power set of V, i.e., the set of all sub-multisets (a multiset is an unordered set where elements can repeat). The *contains*() predicate, likewise defined in ISO 19107, indicates whether a point coordinate lies inside a geometric object. For example, a point cloud may contain more than one value for a given point; the evaluation function will return the multiset of these values for that point. The same holds for curves, surfaces, and solids which all may overlap.

A. Coordinates and Coordinate Reference Systems

The n-D region which a coverage domain spans (we avoid the mathematical term "space" because coverage axes can span more than physical space) is built from n>0 axes. Consequently, point coordinates form an n-tuple where the i^{th} component is taken from the i^{th} axis a_i. The ordered list of axes defines the function domain, described through a Coordinate Reference System (CRS).

Handling of coordinates is normatively established in the ISO 19111:2019 standard [13] whose use is also mandated by 19123-1. Conveniently, beyond geodetic CRSs 19111:2019 also opens the door for further axes and CRSs, as well as combining CRSs. One example for this is image timeseries where a horizontal CRS (contributing two axes) is combined with a 1-D CRS (adding one further axis) into a 3-D CRS. With the OGC CRS shorthand notation the World Geodetic System 1984 (WGS84) CRS [*EPSG*:*4326*] and datetime CRS [*OGC*:*AnsiDate*] get combined as ordered list [*EPSG*:*4326*],[*OGC*:*AnsiDate*].

More details about CRS syntax and handling are specified in the concretization standard 19123-2.

B. Coverage Structures

ISO 19123-1 defines the basic coverage components domain set, range set, and range type:

- *Domain set*: "where are values available?" Points for which values are stored are called *direct positions*.
- *Range set*: "what is the value at a particular position?" Such values consist of records with one or more components (atomic, such as in grayscale images, or composite structures such as color images).
- *Range type*: "what do these values mean?" This describes the semantics for each range value record component (also known as bands / channels / variables).
- *Metadata*: "what else do we know about this coverage?" This item is a black box which literally can be anything, not understood by the coverage but duly transported.

19123-1 does not hardwire the above structure. Rather, several organization schemes are provided:

- by domain and range, plus a mapping between them;
- as a set of direct position / value pairs;
- partitioning of the coverage into sub-coverages.

We discuss each alternative in turn.

The domain/range separation follows directly from the structuring in Figure 2. The advantage is that the domain representation can be chosen independently, which is very important particularly with grid coverages where a detailed structure with several variants is required. On the other hand, the connection between direct positions and values is lost and needs to be established separately. Typically, such a mapping is done through sequence rules inside the coverage function structure defining the correspondences between (implicitly) enumerated direct positions and the simple sequence of values in the range set are established.

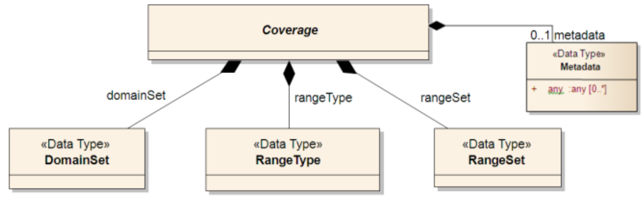


Figure 2. Basic building blocks of a coverage.

 ISBN: 978-1-68558-269-2

The position/value pair approach is attractive whenever the geometry and its associated value are used in conjunction. This is often the case, for example with point clouds. On gridded data, on the other hand, many algorithms work without reference to the geographic coordinates of the pixels, and hence can very efficiently iterate over the values only, disregarding the domain set.

Partitioning can be seen as a generalization of the position / value pair approach where not single pairs, but sets of such pairs are built. Every partition forms a complete, self-contained coverage, and all partitions together must be non-overlapping and contiguous without “holes”. Partitioning schemes are common for splitting large coverages (i.e., “Big Data” files) into smaller “tiles” or “chunks”. In [2], a method for user-invisible flexible partitioning of datacubes is introduced.

C. Coverage Function

Historically, in the coverage definition of the Geography Markup Language (GML) [15], an alternative was foreseen for defining the coverage function analytically. This has never been detailed, GML only vaguely mentions that the Mathematics Markup Language (MathML) might be used. Today, the coverage function is mostly used for describing the internal range set array sequencing through its *sequence rule* subitem.

D. Domain Set

The coverage domain describes for which positions in the coverage’s multi-dimensional space values are available, in other words: where evaluation of the coverage function is defined. Within this multi-dimensional space defined by the domain’s CRS and the bounding box extent, the coverage domain contains a set of geometric objects which together determine the direct positions, i.e., the locations in this space where the coverage offers a value. This description can be given through direct enumeration of the direct positions (example: point clouds) or through containment descriptions (example: areas and volumes), or some other mechanism (example: Ground Control Points in sensor models). The coverage’s “extent” gives a bounding box – i.e., lower and upper bounds along every coordinate axis – within which all its direct positions are located. A quick overview on the footprint of the coverage can be obtained through the coverage envelope.

Coverage coordinates are defined through a single CRS which defines all axes, using ISO 19111:2019. Each axis is described by an axis name, a Boolean axis direction (*true* for positive direction along the axis, *false* for inverse direction), a unit of measure, and a (possibly empty) set of interpolation methods applicable along this axis. As discussed, axes can be of spatial, temporal, or abstract (in ISO 19111:2019: “parametric”) nature.

Note again that this does not yet define a concrete data structure; many different incarnations are possible ultimately carrying the same information. For example, a concrete implementation schema may choose to not define interpolation methods always per axis, but may group several axes – such as Lat and Lon – into a single description.

E. Envelope

A typical first step when shaking hands with a coverage is to ask about its region covered, i.e., its axes and extent along each axis. This information is available in the domain set: By determining the minimum and maximum of point coordinates for each component, the overall extent of the region along each axis is determined. These boundaries determine an axis-parallel minimum bounding box, or *bbox*.

While it is possible to obtain this information from the coverage domain set, it is not straightforward: the n-dimensional domain is described through n axes, possibly of different types, and in some without explicit indication of the lower and upper bounds. Additionally, the domain might employ a CRS different from the desired one. For example, the European Terrestrial Reference System 1989 (ETRS89) system used in Europe consists of 60 different Universal Transverse Mercator (UTM) zones whereas a US GIS may want to see all data in the single WGS84.

The envelope concept provides a shortcut to such information. It contains the bbox of the coverage in a CRS which, for the users’ convenience, can be different from the domain set CRS (as long as a conversion exists between envelope and domain CRS). There is no need for the envelope to be minimal, although it should get as close as possible to the coverage footprint.

F. Range Set and Range Type

Range values listed must adhere to the definition given in the coverage’s range type, following a dynamic typing approach. The range values can be scalar or a record. For simplicity, more involved structures – such as variable-length lists, arrays, graphs, etc. – are not supported in order to keep implementations simple in this respect.

For example, a coverage might assign to each direct position in a county the temperature, pressure, humidity, and wind velocity components *u* and *v*, at a specific time, at that point. The coverage then maps every direct position in the county to a record of these components. The coverage range type, therefore, is a record of these components, each of its individual type.

Type information goes beyond the mere data type as in programming languages. Essential extra information is provided, in particular: Data type; unit of measure; null values, if any. For example, RGB images might have as their range type a record consisting of three components *red*, *green*, and *blue* (in that order), each of them of type unsigned 8-bit integer with unit Watt per square meter – in Unified Code for Units of Measure (UCUM) syntax: *W.m-2* – and no null values. The 19123-2 concretization of ISO 19123-1 adds further details.

G. Interpolation

Having space and time axes, a coverage is a finite, discretized representation of some typically infinite, continuous phenomenon. Digital representations of such fields, therefore, must find appropriate data structures to represent the infinity of points by a finite data volume. Obviously, it is desirable that even positions can be queried for which no value is stored – typically, between direct positions. The general approach is to store a finite number of “representative” points with their values alongside with rules how to derive values at further points.

Under certain conditions, such values can be derived algorithmically through interpolation. Hence, direct positions plus interpolation can emulate the continuous nature of the original phenomenon. Many interpolation methods are known for such purposes, obviously the technically appropriate method has to be chosen carefully to remain sufficiently close to the original.

The interpolation applicable is co-determined by the range type. For example, radiometry data, such as hyperspectral satellite imagery, is normally amenable to linear, quadratic, and cubic interpolation due to the continuous nature of the radiation measured. Categorial data like land use, on the other hand, allow only nearest-neighbour interpolation – the average of street and building does not make sense. Further particularities can have an impact, like the lack of direct positions; kriging is a family of special interpolations used in particular in geophysics.

In summary, interpolation is determined by both domain and range of the coverage function:

- The coverage axis. For example, atmospheric linear interpolation may be fine in Latitude and Longitude, but not vertically when measured in pressure levels. Also, time axis behavior may need to be considered separately. Index axes, finally, with their integer coordinates, do not even allow for addressing fractional coordinates. Within one and the same coverage, different interpolations may apply along different axes.
- The range type (possibly individually for each record element). For example, categorial data (like land use) only allow nearest-neighbour interpolation whereas radiometry etc. also allow linear interpolation.

The coverage standard guides application of interpolation, but does not itself define interpolation methods; these are rather taken from ISO 19107. Only for the reader’s convenience, ISO 19123-1 Annex B addresses interpolation in a non-normative way.

Notably, the abstract coverage concept allows only one interpolation. The reason is that interpolation is a consequential of the physical field structure emulated by the coverage, and different interpolation yields different in between values so represent different fields. For practical reasons – to avoid duplicating Big Data – in 19123-2 CIS a set of “allowed interpolation methods” is foreseen.

A further complication may be the applicability of interpolation around a direct position. Naively, any position between two adjacent direct positions can be queried, and interpolation (if any) will yield a range value. However, being “too far away” from any direct positions, when the neighboring direct positions happen to be far apart from each other, might be to “unsafe” and so interpolation may be forbidden. The concept of a *region of validity* around direct positions captures this, as first introduced for the time axis [5] and implemented in the rasdaman datacube engine. See [6] for future-directed concepts.

Based on these concepts, the original distinction of 19123:2005 into discrete and continuous coverages can be grasped exactly: An axis is called *discrete* if every possible interval with finite bounds describes a finite set of values, otherwise (when interpolation is enabled) such an axis is called *continuous*. A coverage is called *discrete* if its axis list contains only discrete axes. A coverage is called *continuous* if its axis list contains at least one continuous axis. Technically, a continuous coverage is a discrete coverage which can be interpolated.

H. Coverage Classification

The coverage concept in ISO 19123-1:2023 defines a series of different approaches to establish digital structures for spatio-temporally varying phenomena. The idea is to exploit additional knowledge that may exist about the phenomenon. For example, if point values measured sit on a grid (aka grid or raster coverage) rather than arbitrarily in space (aka point clouds) then Computer Science knows specific, very efficient methods to exploit this knowledge.

Following this line, the standard classifies coverage regions into features – points, curves, surfaces, or solids – with potentially additional conditions imposed such as a grid lineup. To keep coverage handling tractable in implementation, only one kind of feature is allowed in any given coverage. This gives a natural classification of coverage, sorted along the topological dimensions of its elements: 0-D point, 1-D line, 2-D surface, and 3-D solid coverages. This is mirrored by the coverage types in Clause 6 onwards in ISO 19123-1:2019 in multi-point, multi-curve, multi-surface, and multi-solid coverage.

A *multi-point* coverage is a coverage consisting of a collection of 0-D points. As points may coincide, there can be more than one value correspond to a given direct position, therefore the evaluation returns a multi-set of values with possibly more than one value.

A *multi-curve* coverage resembles a set of geometric objects of the ISO 19107 type *CurveData*. Curves defined there encompass a wide range, from polygon strings to splines. AIS worldwide ship tracking system trajectories represent an example of multi-curve coverages. Trajectories may intersect, hence *evaluate*() may deliver more than one trajectories as values.

A *multi-surface* coverage is a coverage consisting of a collection of surfaces. The feature type used is given by the ISO 19107 geometric object type *SurfaceData*. Such surfaces are described through bounding curves which in turn are delimited by start and end points. A typical example for a multi-surface coverage is an iso-surface set.

A *multi-solid* coverage consists of a collection of solids, modeled through ISO 19107 *SolidData* which adopts a Boundary Representation where solids are bounded by surfaces delimited by curves delimited by points.

I. Grid Coverages

A grid coverage is a special case of multi-point coverage: all direct positions must sit on a grid. As the grid structure is of prime practical importance, we unfold it separately.

Mathematically, an n-D grid is the cross product over the admissible coordinates of each contributing axis. For some n>0 let $A = (a_1, \ldots, a_n)$ be a finite ordered set of axes where each axis $a_i = \{v_{i,1},\ldots, v_{i,mi}\}$ is an ordered set of $m_i>0$ values inducing a grid $G = a_1\times\ldots\times a_n$. G can be interpreted as a set of coordinates yielding the direct positions, $G = \{ (x_1, \ldots, x_n) \mid x_i \in a_i \text{ for } 1 \leq i \leq n \}$.

Such a grid consists of points only. These points are aligned in a special way, and we often like to draw lines between neighboring points so that the alignment becomes easier to see. However, these lines are artifacts and not part of the coverage grid. Notably, the gridded nature does not affect the CRS in any way – the grid is just about constraints on the coordinates.

Geometrically, grids generally can be constructed based on triangles, rectangles, or hexagons (meaning: the grid points can be aligned so that, *would* they be connected, we *would* see such geometric shapes). In the context of ISO 19123-1, rectangular grids are modeled through grid coverages, hexagonal grids can be mapped to grid coverages, and triangular grids are modeled through meshes, i.e., multi-surface or multi-solid coverages. In the sequel, for simplicity the term “grid” is understood as a rectangular grid.

Intuitively speaking, in a coverage grid, every direct position (except at the rim) has exactly one immediate neighbor with a lower coordinate and exactly one immediate neighbor with a higher coordinate along each axis (Figure 3. This neighborhood establishes the grid topology; the grid geometry is determined by the concrete coordinates, which in turn are described by the axis types.

The grid alignment constraint also has a further consequence: As it is not possible any longer that two points coincide, there will be always one range value per direct position, and we can simplify the *evaluate*() function from a value set to a single value:

$evaluate_C: D \rightarrow V,\ evaluate_C(p) = v$

J. Regular and Non-Regular Grids

In general, rectangular grids do not need to have an equidistant spacing between the direct positions. Figure 4 and Figure 5, taken from the standard document, illustrate some cases of regular and irregular grids. A grid can be regular along some axes but irregular along others, as Figure 5 shows. In particular, when grid connections are drawn as curved lines, this should not be interpreted as reality.

The grid concept can be generalized to the situation that n-D grids can be embedded in some $(n+m)$-D space for some $m > 0$. Actually, Figure 5 (c) models such a situation where a 2D grid is warped in 3D space.

K. Grid Axis Types

ISO 19123-1 categorizes the coverage grid domain by its individual axes, allowing free combinations such as regular spatial with irregular temporal axes. Notably, this axis *classification* establishes several ways to describe the coordinates of the direct positions, not the grid CRS which contains the axis *definitions*.

Every axis has one of the following axis types: index, regular, irregular, warped, and (sensor) model.

An *index axis* is a 1D unit-less axis (in ISO 19111:2019 named “Cartesian axes”); there is no georeference, and admissible coordinates are at discrete, integer positions only. The corresponding CRS is *Index1D* for a single axis, and *Index2D* etc. for a multi-axis setup. For two lower and upper bounds *lo* and *hi* with $lo, hi \in \mathbf{Z}$ and $lo \leq hi$, the direct positions are taken from the closed interval $S = \{ x \in \mathbf{Z} \mid lo \leq x \leq hi \}$.The bounds, at the same time, constitute the bbox along this axis.

A *regular axis* has an equi-distant spacing like an index axis, but is continuous and not constrained to integer positions and distances. It can be georeferenced, i.e., it can have a spatial or temporal (or other) semantics attached, given by its CRS. It can be described conveniently by lower and upper bound plus resolution.

An *irregular axis* lists (possibly georeferenced) positions $P = \{p_1,\ldots, p_n\} \subseteq C$ explicitly where C denotes the coordinate value set defined for this axis in the CRS. Direct positions exist for every coordinate tuple where the coordinate value of the irregular axis is from P.

A *displacement axis nest* (or *warped nest*) is a set of possibly georeferenced axes forming a subset of the CRS’s axes. Direct positions have maximum freedom of location, the only rule being that coordinates along each participating axis remain ordered and no duplicate coordinates appear. Direct positions are given by the coordinate tuples where the coordinate of each axis participating in the displacement axis nest is in the coordinate value set of this axis.

By combining all the above axis types freely, any type of grid shape can be modeled. The list of possible axis types in the standard is not exhaustive, some standard or application may define their own additional axis types.

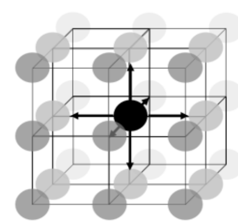
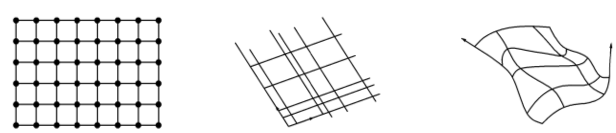
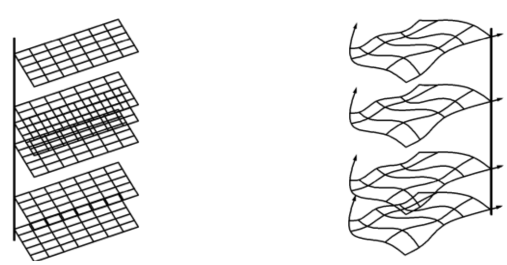


Figure 3. Multi-dimensional neighbourhood in a grid [10].



(a) (b) (c)

Figure 4. Sample regular 2-D grid (a), 2-D irregular grid (b), 2-D warped nest grid (c) [10].



(a) (b)

Figure 5. Sample 3-D x/y/t grid representing the combination of regular Lat/Long with irregular time (a) and warped nest with irregular time (b), time axis running vertically [10].

Obsoleted ISO 19123:2005 differentiates on grid level distinguishing only rectified and referenceable grid coverage. Based on the above grid construction mechanisms, these terms can be defined precisely:

- A *rectified grid coverage* is a grid coverage where every axis is either an index axis or a regular axis;
- A *referenceable grid coverage* is a grid coverage where at least one axis is neither index nor regular axis.

L. Grid Cells

Inspired by the Computer Science term of "array cells" – storage locations in memory for the values, lined up in sequence – geo informatics also has a common notion of "grid cells", however with different understanding. In a grid cell view, the imaginary lines suggest to be boundaries of an area which suddenly becomes the first-class citizen. Consequently, questions arise like "is the real cell location at the direct position or rather between the direct positions, in the center of the cell?" and "is the cell extent still a point like the direct position, or is it an area now?"

This is captured by the commonly used, yet not clearly defined distinctions *pixel-in-corner* versus *pixel-in-center* on the one hand and *pixel-is-point* versus *pixel-is-area* on the other hand.

These questions will be addressed in a forthcoming paper, aiming at a comprehensive conceptual treatment.

III. COVERAGE IMPLEMENTATION SCHEMA

We next address the coverage concretization standard, ISO 19123-2 [11], known as Coverage Implementation Schema (CIS). CIS is a compliant standardization target of ISO 19123-1:2023, meaning: it relies on the concepts, terms, definitions, and interfaces of the abstract data model to establish a logical schema expressed in the Unified Modeling Language (UML) implementing the interfaces defined there. Additionally, this document defines several format encodings for the single logical schema.

Current ISO 19123-2:2018 was adopted from OGC CIS 1.0; integration of OGC CIS 1.1 [27] is under work as a version update. In the sequel, we introduce the latest, yet unpublished draft (named CIS for short), thereby providing the most up-to-date information to the public while work is still in progress.

OGC CIS 1.1 does not supersede, but extend OGC CIS 1.0. When integrating both into a self-contained ISO 19123-2 a specific structure had to be found for the combined document because both differ in places due to historical reasons. With a similar approach as in 19123-1, the CIS 1.1 coverage classes have been put into the specification body while isolating the legacy – consisting of the rectified and referenceable grid coverages – in a separate annex.

One important reason for fencing CIS 1.0 and 1.1 is due to the GML legacy. The GML 3.2.1 coverage structure [15] is both overly complicated and too restrictive. The complication comes from a particular modeling style of GML which might be academically justified but in practice almost duplicates the number of structuring elements in the GML encoding. The most important of the restrictions is due to the coordinate types which normatively are fixed to numbers in GML. However, in today's timeseries and datacube world temporal axes require date and time stamps, such as "2025-01-25" – nobody wants to count seconds since January 1st 1970. All communities made clear that support for convenient calendar and time syntax is an absolute must. Still, despite manifold requests and discussion the GML working group was not willing to extend GML with strings. Therefore, CIS 1.1 carefully deviates from GML to allow any type of coordinates.

Additionally, the domain set description in the CIS 1.1 *GeneralGridCoverage* has been made more straightforward.

In a nutshell, the main changes of CIS over its predecessor version ISO 19123-2:2018 are as follows:

- CIS has been adjusted to ISO 19123-1 in terminology and concept use, with a clear focus and separation into logical level (UML structures) and physical level.
- All CIS 1.1 coverage classes are adopted unchanged. Legacy grid coverage classes *RectifiedGridCoverage* and *ReferenceableGridCoverage* (the latter from a separately adopted OGC standard [35]) have been retained, but moved into a separate (normative) Annex B. These two types are legacy and will be deprecated in the next version – anyway, *GeneralGridCoverage* can model these cases while simpler in structure.
  Technically, gridded coverages still consist of an *n*-D matrix (mathematically: tensor), ornamented with extra information realizing the spatio-temporal semantics.
- The JSON encoding of CIS 1.1 has been reworked to comply with modern JSON Schema.
- Due to resource reasons, the Resource Data Framework (RDF) encoding present in CIS 1.1 has not been included at this time and is left for future work.

Realizing the structuring opportunities of 19123-1:2023 CIS likewise offers several structuring variants: a separation of domain and range, partitioning into sub-coverages, and direct enumeration of position/value pairs (sometimes also called "geometry / value pairs" or "interleaved representation"). In this overview, we limit ourselves to the very common domain/range representation.

B. Coordinates and Coordinate Reference Systems

Direct positions are expressed as coordinate tuples, as laid down in ISO 19123-1. Coordinate values are of data type *string* as they must accommodate data types as diverse as numbers (such as 1.23 degrees or 500 nm), dates and times (such as "2016-03-08T11:23Z"), categorial values (such as "orange", "apple"), and possibly more.

Similarly, resolution specifications are of type string as they have to accommodate, e.g., "1.23" for degrees or meters and "PT2h" for a 2-hour duration. As per ISO 19111:2019, any coordinate representation scheme must convey some total ordering so that expressions like "lowerBound ≤ upperBound" are valid for any axis.

We briefly focus on date and time coordinates as these convey a more involved syntax. The ISO 19108:2002 standard [16] applies here which defines the date and time syntax used, such as "2023-01-01T10:15:22.345Z" and "2023-01-01T00:00:00.000CET". Note the time zone identifiers, "Z" (for Zulu time aka UTC) and "CET". Such timestamps are called “fully qualified”; shorter time strings with different temporal resolution are possible, such as "2023-01-01" and "2023". The basis for date and time is one basic time CRS counting in seconds. On top of this, calendar CRSs are built such as *GregorianDateTime* (following the syntax sketched above), *UnixTime*, and *Chronometric-GeologicTime*.

Several vertical CRSs are available in the OGC registry. What still has to be added are proxies such as pressure altitude (measured in *hPa* or *psi*) for altitude in the atmosphere. Their description likely is possible through parametric CRSs foreseen in ISO 19111:2019.

Such coverage axes are defined by the coverage CRS as laid down in ISO 19111:2019 [13]. Any combination of spatial, temporal, and “abstract” (i.e., non-spatio / temporal) axes is possible. This coverage CRS – its so-called *native CRS*, in which data are stored in the coverage – is a single *n*-D CRS for the *n*-D coverage. (This is an important difference to other spatio-temporal data standards in OGC which split CRS components over several places, an approach which is not only more difficult to oversee but also comes with significant conceptual restrictions.)

OGC several years back has resolved that CRSs are to be expressed through URLs, such as the following for WGS84, which has EPSG [36] code 4326:

https://www.opengis.net/def/crs/EPSG/0/4326

In the *crs-compose*/ branch, component CRSs can be added constituting a concatenation as per ISO 19111:2019. For example, a 3-D t/x/y CRS can be built from ETRS89 LAEA and date/time by concatenating two CRS URLs:

https://www.opengis.net/def/crs-compound?
1=https://www.opengis.net/def/crs/OGC/0/AnsiDate&
2=https://www.opengis.net/def/crs/EPSG/0/3035

Such URLs can be “resolved” using the OGC CRS Resolver service [32] which returns the XML-encoded definition of the CRS.

These long, hard-to-read URLs mostly are geared towards machine consumption – nevertheless, they were felt unwieldy, and so the rasdaman team at some time suggested a bracket notation as shorthand. Meanwhile these shortcuts are adopted by the OGC Naming Authority and permitted as alternatives to the CRS URLs. Rules are simple:

- A non-composite CRS URL of pattern https://www.opengis.net/def/crs/{*authority*}/{*version*}/{*id*} is identical to the shorthand [{*authority*}:{*id*}]
  Version number is 0 by definition, interpreted as "latest available". For example, [*EPSG*:*4326*] expands to https://www.opengis.net/def/crs/EPSG/0/4326
- A composite CRS URL is translated into a comma-separated sequence of the component CRSs, each of which is transcribed individually as per the rule above. For example, [*EPSG*:*4326*],[*OGC*:*AnsiDate*] is equivalent to the long version
  https://www.opengis.net/def/crs-compound?
  1=https://www.opengis.net/def/crs/OGC/0/AnsiDate&
  2=https://www.opengis.net/def/crs/EPSG/0/4326

Such CRS shorthand can be used, e.g., in the *srsName* attribute of a coverage domain set (see below), like:

srsname="[EPSG:4326],[OGC:AnsiDate]"

Note, however, that not all coverage implementations necessarily implement this feature; notably, the rasdaman WCS reference implementation does support it.

Based on this CRS infrastructure, we can define *n*-tuple coordinates for direct positions in coverages. Thanks to the generalization of CIS 1.1 and the liberation from GML restrictions, coordinates can be numeric and non-numeric alike.

C. *Coverage Domain Set*

The coverage domain set specializes into specific structures for multi-point, grid, multi-curve, multi-surface, and multi-solid domain set specifications as discussed earlier. All have in common, though, the *srsName* attribute holding the CRS of the coverage using either URL or bracket notation. In attribute *axisLabels*, the list of axis names in the CRS is provided in proper order, whitespace separated. These axis names are used inside the coverage for axis identification in the domain set’s axis list. In attribute *uomLabels* the unit of measure is indicated for each axis in a whitespace-separated list in proper axis order. Best practice is to use UCUM notation [38] such as “m”, “ft”, “yr”, etc.

In grid coverages, the *GeneralGrid* structure inside the *DomainSet* serves to span the *n*-D raster grid. For each axis its type is defined which mirrors the 19123-1 definitions.

An *IndexAxis* constitutes the simplest axis type, with only integer coordinates allowed. No resolution and no unit of measure are required.

A *regular axis* employs as coordinates any totally ordered value set, such as numbers and date/time strings. Additionally, the unit of measure – recommended: UCUM – plus the (constant) resolution need to be kept.

An *irregular axis* is like a regular one in that it can use any totally ordered value set for coordinates, with the unit of measure to be indicated. The coordinates contributing the direct positions are enumerated explicitly.

We omit the further axes types – irregular correlated grid axes (also called *displacement axis nest* or *warped nest*) and *transformation model* – to avoid undue complexity in this overview paper.

D. *Coverage Range Set and Range Type*

The range set usually forms the by far largest part of the coverage in terms of its storage footprint. Therefore, this part is designed as compact as ever possible, with no redundancy – the structure simply resembles an ordered list of values. It is essential, therefore, to have a linearization rule establishing a clear correlation between the multi-dimensional direct positions and the 1-D value sequence. The default row major / left-to-right sequencing rule can be overridden in the *sequenceRule* part of *CoverageFunction*.

The range type adds technical metadata required for a program to interpret the coverage range values correctly. CIS makes use of the OGC Sensor Web Enablement (SWE) Common [25] *DataRecord*. This ensures that the semantics from upstream sensor acquisitions into downstream services (like WCS) is carried over losslessly. Each range value can be a record characterized by component field name, unit of measure, and a characterization into Quantity, Count, or

 ISBN: 978-1-68558-269-2

Category. Further optional parts include nil (null) value list, definition (a URL pointing to a human-readable definition), and further more.

Besides *DataRecord*, there is an optional list of interpolation methods applicable. Common interpolation methods include *nearest-neighbor*, *linear*, *quadratic*, *cubic*, *barycentric*, and more. Interpolation is tightly connected with the region-of-validity concept, something to be reflected in subsequent standardization progress once there are conclusive results from the ongoing research [5][6].

E. Metadata

The metadata slot is as defined abstractly before: some byte string without further semantics known to the coverage. Use of this slot is manifold: To enhance the coverage information; to provide further domain-specific information; to create profiles, such as EU INSPIRE metadata [20].

F. Coverage Encodings

Many encoding formats are in active use for coverages in practice. Several of those are already standardized, such as GeoTIFF, NetCDF, GRIB2, and JPEG2000 – see the list at [31]. XML and JSON encodings are already contained in OGC CIS 1.1 [27] as separate conformance classes.

The XML encoding has a strong legacy from GML [10] to which it was aligned at the heydays of XML use. GML coverages came with several constraints (such as numerical coordinates only), and so a cautious liberation of GML was started with OGC CIS 1.1 allowing date / time strings and simplifying the structure.

Further, OGC CIS 1.1 added a conformance class for JSON. While reworking this into the new version of 19123-2 this was reshaped to match with current technology, in particular: JSON Schema [21].

The ASCII formats XML and JSON are "informationally complete" by containing all of the coverage information defined, but they not efficient in particular for voluminous data. Efficient binary formats, on the other hand, tend to grasp only part of the coverage information. For an encoding which is both informationally complete and storage efficient the multi-part conformance class was added. It defines a container which, as first item, contains an overall coverage description in some well-known complete format like XML or JSON. Instead of the storage-heavy parts – typically the range set – a reference is provided to one or more files also stored in the container. These further parts can be in any well-known format, typically in a compact binary encoding.

IV. Coverage Wrangling Standards

While this paper focuses on the coverage data structure, we still discuss briefly the corresponding service standards. The direct companion service standard to the coverage data standards is the OGC *Web Coverage Service* (WCS) which offers versatile extraction, conversion, analysis, and fusion on general multi-dimensional datasets [31]. Part of the modular WCS suite is the *Web Coverage Processing Service* (WCPS) [3][26], a geo datacube analytics language built for server-side evaluation. WCS is supported by manifold implementations [30], such as Oracle, Hexagon, GeoServer, ESRI, and rasdaman.

For map visualization, OGC Web Map Service (WMS) and Web Map Tiling Service (WMTS) are available. As opposed to WCS, these are specialized on 2D map rendering of datasets with two horizontal axes. WMS returns color pixels (like hill shading), a WCS delivers the original data (like height in feet) in a way that allows further processing.

For WCS and WMTS, rasdaman is official OGC Reference Implementation.

Given that coverages are "Big Data", they typically are "too big to download", hence processing requires "shipping code to data". From a service provider perspective, unguarded acceptance of programming language code is unsafe; from a user perspective, coding requires extra skills making exploitation infeasible for non-experts and time-consuming for experts. Therefore, OGC, ISO, and INSPIRE have adopted the dedicated datacube analytics language Web Coverage Processing Service (WCPS) [3][12]. This language defines expressions on coverages which evaluate to ordered lists of either coverages or scalars (whereby "scalar" here is used as a summary term of all data structures that are not coverages). Like the SQL data analytics language, WCPS is "safe in evaluation": every query is guaranteed to terminate in finite time, as opposed to programming languages like Python where such a guarantee is not possible.

We present WCPS through some examples illustrating basic mechanisms; see also the WCPS tutorial on EarthServer [22] and the ChatCUBE WCPS query assistant [23]. A forthcoming paper, updating the original WCPS 1.0 overview [3], will address WCPS 1.1 in detail.

- "Retrieve coverages A, B, and C in GeoTIFF":
```
for $c in ( A, B, C )
return encode( $c, "image/tiff" )
```
- "Apply mask M to coverage A, B, and C" (fusion):
```
for $s   in ( A, B, C ), $m in ( M )
return encode( $s * $m, "image/tiff" )
```
- "Create 3D x/y/t coverage from input stream $1":
```
for $t in ( TemperatureCube )
return encode(
  coverage MySatelliteDatacube
  domain
    crs "EPSG:4326+OGC:unixTime" with
    Lat  regular (10:30) resolution 0.5
      interpolation linear,
    Lon regular (10:30) resolution 0.5
      interpolation linear,
    Date irregular ( "2017-01-01", "2017-02-01",
                     "2017-07-01", "2017-11-01" )
  range type panchromatic: integer
  range decode( $1 ),
  "netcdf"
)
```
- "Timeseries of temperature average over Berlin":
```
for $t in ( TemperatureCube )
return encode(
   avg( $t[ Lat(52.51: 52.53), Lon(13:39:13.41) ] ),
   "json"
)
```


 ISBN: 978-1-68558-269-2

- "Absolute of wind speed":
```
for $w in ( WindCube )
return encode(
   sqrt( $w.u * $w.u + $w.v * $w.v ),
   "netcdf"
)
```
- "Logarithm of intergalactic matter temperature ":
```
for $c in ( UniverseTemperature )
return encode(
   switch
      case $temp > 0 return log( $temp )
      default        return 0,
   "netcdf"
)
```

The syntax of WCPS tentatively is aligned with XQuery – a majority of geo metadata are stored in XML, so naturally queried with XPath / XQuery. This allows for an integration of the two languages into a seamless data / metadata continuum. Furthermore, XQuery is also suited for querying JSON structures, so future oriented.

V. Related Standards

The coverage standards, aligned between ISO and OGC, are generally accepted and widely implemented. In this section we inspect related standards.

With SQL Part 15 (*Multi-Dimensional Arrays*, *MDA*) [9], ISO has added multi-dimensional arrays to the relational model. MDA defines how attribute values can be arrays of arbitrary extent and number of dimensions, including operational support in the SQL query language. These arrays are domain-agnostic and not aware of spatial nor temporal semantics. The OGC/ISO *Web Coverage Processing Service* (WCPS) language [3][12] is different in that (i) it adopts an XQuery syntax flavor to be better aligned with the many geo metadata stored worldwide and (ii) is aware of space and time, knowing, e.g., about regular and irregular grids. However, the operational semantics is the same as SQL/MDA, except that WCPS is space/time semantics aware. This is exploited, for example, in the rasdaman array database system where WCPS queries internally get translated, with the help of geo-specific metadata, into SQL/MDA style queries which ultimately are executed in the federated engine [9].

CoverageJSON [34] is an OGC community standard for datacubes. Despite its name it is not the JSON encoding of coverages, but an incompatible variant – a "hijacking" of the normatively defined name "coverage".

W3C QB4ST [1] establishes a datacube ontology, expressed in Resource Data Framework (RDF) syntax and queryable through the RDF query language, SPARQL. QB4ST only addresses datacube metadata, but not the "payload" itself. While an interesting approach in itself, with a potential to bridge into the Semantic Web world, QB4ST likewise is not aligned with the coverage standards.

While focus here is on the coverage data model we briefly address service APIs. The first and foremost coverage service standard is the *Web Coverage Service* (WCS). In its core, it offers only subset extraction and format encoding so as to keep the implementation hurdle as low as possible. A series of optional extensions adds further functionality. Particularly noteworthy is WCPS, a high-level geo datacube query language.

Further relevant standards include *Web Map Service* (WMS) for map visualization. WMS and WCS differ in that WMS focuses on map visualization, hence returns colors (such as color shading for elevation levels) whereas WCS delivers the true data (such as elevation), suitable for further processing and analytics by tools.

Some further standards, such as *Environmental Data Retrieval* (EDR) [24], use (incompatible) CoverageJSON.

OAPI-Coverages offer access to coverages based on OpenAPI technology and http. Functionality is mostly parallel to WCS. The specification is draft since about 2018, but still incomplete, with random changes, without a comprehensive example set nor a test suite, and altogether not suspected to become OGC standard in the near future.

Another recent OGC activity has started work on a *GeoDataCube* API which itself consists of two incompatible API definitions, openEO and OAPI-Processes. It is likewise an early-stage draft under discussion.

The European legal framework for a common spatial data infrastructure, *INSPIRE*, relies on the OGC coverage standards, including WCS and WCPS [20].

VI. Conclusion

Standardization not only fosters interoperability, but also offers guidance to implementers, thereby accelerating development cycles. Conversely, scientific and technological progress in the understanding of generation, management, and use of coverage structures nurtures the standards continuously. Coverages have matured in concepts and implementation, culminating in CIS 1.1. The integration of both is to become the next version of ISO 19123-2.

This paper provides a lookout on this new standard synoptically on three levels of abstraction: the concepts and terminology of ISO 19123-1, the logical-level coverage data model of ISO 19123-2 which currently is under adoption vote, and the physical (encoding) level of ISO 19123-2 providing XML and (revised) JSON support, in addition to the existing binary coverage formats. The first ISO vote ("ballot") was finished with only minor comments. These have been worked in, making the specification ready for the next stage ballot (Draft International Standard, DIS). From DIS status onwards only editorial changes will be allowed. Altogether, the document can be considered quite stable.

Coverage data and service standards have an immense impact on Big Geo Data, in particular datacubes – examples include 1-D sensor timeseries; 2-D satellite, airborne drone and underwater data, on Earth or on planetary bodies; 3-D x/y/t image timeseries over all these; 3-D x/y/z geophysical data, such as with oil, gas, and water exploration; 4-D x/y/z/t atmospheric and ocean data; and general n-D statistical datacubes. These few examples may illustrate the importance of coverages for geo data in science and industry.

The contribution, therefore, aims at spreading information about coverages in general and datacubes in particular, and conversely solicits feedback by the community into standardization.

ACKNOWLEDGEMENT

The rasdaman team is doing brilliant work in shaping a datacube engine implementing the standards comprehensively. The author is grateful for the many intense discussions in OGC. Work in part was supported by NATO SPS Cube4EnvSec, EU FAIRiCUBE and EU EFRE FAIRgeo.

REFERENCES

[1] R. Atkinson, "*QB4ST: RDF Data Cube extensions for spatio-temporal components"*. W3C Working Group Note / OGC Document, pp. 16-142, 2017, https://www.w3.org/TR/qb4st, [retrieved: 04, 2025].

[2] P. Baumann, S. Feyzabadi, C. Jucovschi, "*Putting Pixels in Place: A Storage Language for Scientific Data"*. Proc. IEEE ICDM Workshop on Spatial and Spatiotemporal Data Mining (SSTDM'10), December 14, 2010, Sydney, Australia, pp. 194 – 2012010.

[3] P. Baumann, "*The OGC Web Coverage Processing Service (WCPS) Standard"*. Geoinformatica, 14(4)2010, pp. 447-479.

[4] P. Baumann, "*A Database Array Algebra for Spatio-Temporal Data and Beyond"*. Intl. Workshop on Next Generation Information Technologies and Systems (NGITS), July 5-7, 1999, Zikhron Yaakov, Israel, LNCS 1649, pp. 173-189

[5] P. Baumann, "*Enhanced Calendar Support for Temporal Datacube Queries"*. Transactions in GIS, 28, pp. 2089-2112, DOI: 10.1111/TGIS.13215.

[6] P. Baumann, "*On the Analysis-Readiness of Spatio-Temporal Earth Data and Suggestions for its Enhancement"*. Environmental Modelling and Software, Volume 176, 2024, DOI: 10.1016/j.envsoft.2024.106017.

[7] A. Dumitru, V. Merticariu, P. Baumann, "*Exploring cloud opportunities from an array database perspective"*. ACM SIGMOD DanaC, 2014, pp. 10:1-10:4.

[8] R. Haag, "*Local Quantum Physics: Fields, Particles, Algebras"*. Springer, 1996.

[9] ISO, "*Information technology - Database languages - SQL - Part 15: Multi-Dimensional Arrays"*, ISO 9075-15:2019.

[10] ISO, "*Geographic information - Schema for coverage geometry and functions - Part 1: Coverage implementation schema"*. IS 19123-1:2023, https://www.iso.org/standard/70743.html, [retrieved: 04, 2025].

[11] ISO, "*Geographic information - Schema for coverage geometry and functions - Part 2: Coverage implementation schema"*. IS 19123-2:2018, https://www.iso.org/obp/ui/#iso:std:iso:19123:-2:ed-1:v1:en, [retrieved: 04, 2025].

[12] ISO, "*Geographic information - Schema for coverage geometry and functions - Part 3: Processing fundamentals"*. ISO 19123-3, https://committee.iso.org/sites/tc211/home/projects/projects---complete-list/iso-19123-3.html, [retrieved: 04, 2025].

[13] ISO, "*Geographic information – Referencing by coordinates"*. IS 19111:2019, https://www.iso.org/standard/74039.html, [retrieved: 04, 2025].

[14] ISO, "*Geographic information - Spatial schema"*. ISO 19107:2019, https://committee.iso.org/sites/tc211/home/projects/projects---complete-list/iso-19107.html, [retrieved: 04, 2025].

[15] ISO, "*Geographic information - Geography Markup Language (GML) Part 2, "Extended schemas and encoding rules"*, ISO 19136-2:2015.

[16] ISO, "*Geographic information - Temporal schema"*. ISO 19108:2002, https://www.iso.org/standard/26013.html, [retrieved: 04, 2025].

[17] C. Jucovschi, P. Baumann, S. Stancu-Mara, "*Speeding up Array Query Processing by Just-In-Time Compilation"*. IEEE Intl. Workshop on Spatial and Spatiotemporal Data Mining (SSTDM), Pisa, Italy, 2008, pp. 408 – 413.

[18] V. Merticariu and P. Baumann, "*Massively Distributed Datacube Processing"*. IEEE Geoscience and Remote Sensing Society (IGARSS), Yokohama, Japan, 2019.

[19] D. Misev and P. Baumann, "*Enhancing Science Support in SQL"*. IEEE Big Data Workshop on Data and Computational Science Technologies for Earth Science Research, Santa Clara, USA, 2015, pp. 2201 – 2204.

[20] N.n., "*INSPIRE Coverages Demystified"*. https://inspire-wcs.eu, [retrieved: 04, 2025].

[21] N.n., "*JSON Schema"*. https://json-schema.org, [retrieved: 04, 2025].

[22] N.n., "*EarthServer Datacube Federation"*. https://earthserver.world, [retrieved: 04, 2025].

[23] N.n., "*Say Hello to Datacubes*. https://ai-cu.be/chatcube, [retrieved: 04, 2025].

[24] OGC, "*OGC API - Environmental Data Retrieval Standard*. Document 19-086r6, http://www.opengis.net/doc/IS/ogcapi-edr-1/1.1, [retrieved: 04, 2025].

[25] OGC, "*SWE Common Data Model Encoding Standard 2.0*. Document 08-094r1, https://portal.ogc.org/files/?artifact_id=41157, [retrieved: 04, 2025].

[26] OGC, "*Web Coverage Processing Service (WCPS) Language Interface Standard*. Version 1.1, OGC Document 08-068r3, https://docs.ogc.org/is/08-068r3/08-068r3.html, [retrieved: 04, 2025].

[27] OGC, "*Coverage Implementation Schema"*, version 1.1.1. OGC 09-146r8, http://docs.opengeospatial.org/is/09-146r8/09-146r8.html, [retrieved: 04, 2025].

[28] OGC, "*Abstract Specification Topic 6.1 – Schema for Coverage Geometry and Functions – Part 1: Fundamentals"*. OGC Document 07-011r2, https://docs.ogc.org/as/07-011r2/07-011r2.pdf , [retrieved: 04, 2025].

[29] OGC, "*Abstract Specification Topic 6.3 – Schema for Coverage Geometry and Functions – Part 3: Processing Fundamentals"*. OGC Document 21-060r2, https://docs.ogc.org/as/21-060r2/21-060r2.pdf, [retrieved: 04, 2025].

[30] OGC, "*Registered WCS Implementations"*. https://portal.ogc.org/public_ogc/compliance/implementing.php?&specid=533, [retrieved: 04, 2025].

[31] OGC, "*Web Coverage Service (WCS)"*. https://www.ogc.org/publications/standard/wcs, [retrieved: 04, 2025].

[32] OGC, "*CRS Definition Resolver"*. https://external.ogc.org/twiki_public/CRSdefinitionResolver/WebHome, [retrieved: 04, 2025].

[33] N.n., "*OGC Spatio-Temporal Coverage / Datacube Standards"*. https://myogc.org/go/coveragesDWG, [retrieved: 04, 2025].

[34] OGC, "*OGC CoverageJSON Community Standard"*. OGC 21-069r2, https://docs.ogc.org/cs/21-069r2/21-069r2.html, [retrieved: 04, 2025].

[35] OGC, "*Coverage Implementation Schema – ReferenceableGridCoverage Extension with Corrigendum"*. OGC 16-083r3, http://docs.opengeospatial.org/is/16-083r3/16-083r3.html, [retrieved: 04, 2025].

[36] OGP, "*EPSG Geodetic Parameter Dataset"*. https://epsg.org

[37] rasdaman team, "*rasdaman"*. https://rasdaman.org, DOI: https://doi.org/10.5281/zenodo.1040170, [retrieved: 04, 2025]

[38] Regenstrief Institute, "*UCUM"*. https://ucum.org, [retrieved: 04, 2025].