OGC® DOCUMENT: 25-042

External identifier of this OGC® document: http://www.opengis.net/doc/PER/T21-GIMI-D204

Open Geospatial Consortium

OGC TESTBED-21 ONTOLOGIES FOR COVERAGE DESCRIPTION REPORT

ENGINEERING REPORT

PUBLISHED

Submission Date: 2026-05-15
Approval Date: 2026-06-04
Publication Date: 2026-08-12
Editor: Simon J D Cox

Note

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

CONTENTS

I OVERVIEW

This Report documents outcomes from OGC Testbed-21 focused on ontologies for coverage description, with an emphasis on imagery and video. The work responds to a growing need for metadata models that support discovery, understanding, and integration of coverage data across heterogeneous systems, while aligning with both existing OGC geospatial standards and sponsor requirements for ontology-based approaches.

II EXECUTIVE SUMMARY

This Testbed-21 Report documents outcomes from OGC Testbed-21 focused on ontologies for coverage description. The activity examines the draft Imagery Domain Ontology (IDO), which is grounded in the Basic Formal Ontology (BFO) and Common Core Ontologies (CCO), and evaluates how it can interoperate with established OGC Standards such as GeoSPARQL, the Coverage Implementation Schema (CIS), SWE Common, and STAC, as well as the candidate GeoDCAT standard. The report identifies tensions between highly axiomatized upper ontologies and the pragmatic, widely deployed geospatial standards that underpin current operational systems.

The main contribution of the work is a proposed Testbed-21 prototype ontology that integrates selected OGC capabilities into the IDO design. Key elements include the use of GeoSPARQL for geometry representation in RDF, the introduction of an explicit 'region of interest' class to provide a clear real-world referent for imagery, and the encapsulation of CIS RangeType definitions using SWE Common JSON within RDF. This approach preserves semantic rigor while enabling reuse of mature standards and existing software ecosystems.

The report demonstrates the feasibility of this approach through detailed models and worked examples, showing how image geometry, pixel-to-ground correspondences, and image cell properties can be consistently described. It also discusses architectural considerations such as encapsulation strategies, the separation of semantics from mathematical geometry, and the implications of expressing complex data structures in RDF.

Based on the Testbed findings, the report recommends that future versions of the IDO adopt GeoSPARQL for spatial information, retain an explicit real-world referent for imagery, continue short-term use of SWE Common JSON for coverage cell properties, and pursue future alignment with sensor and processing models through SSN-compatible process-flow patterns. Additional future work is identified to develop mappings to catalog and access standards such as OGC API – Records, STAC, and CIS.

Overall, the work shows that a hybrid approach—combining a rigorous ontology framework with well-established geospatial standards—is both practical and beneficial, offering a credible path toward interoperable, semantically rich metadata for imagery and other coverage data.

III KEYWORDS

The following are keywords to be used by search engines and document catalogues.

metadata, RDF, JSON-LD, imagery, geospatial, ontology, BFO, CCO, GeoSPARQL, coverage

IV FUTURE OUTLOOK

The Testbed-21 results point to several important directions for future work in ontology-based coverage description and interoperability within the OGC Standards ecosystem. As semantic technologies mature and operational systems increasingly demand richer machine-interpretable metadata, the integration of ontologies with established geospatial standards is expected to become more central rather than experimental. In the near term, further refinement and stabilization of the IDO is anticipated, particularly through stronger alignment with GeoSPARQL as the normative approach for expressing spatial information in RDF. Consolidating this alignment would improve consistency across implementations and lower barriers for adoption by organizations already invested in GeoSPARQL-enabled tooling. Parallel work to formalize the 'region of interest' concept across imagery and other coverage types would help promote conceptual clarity and cross-domain reuse.

Another key area of future development is the treatment of coverage range and cell properties. While short-term encapsulation of SWE Common JSON within RDF has proven practical, longer-term evolution may explore tighter semantic representations or profiles that balance expressiveness, performance, and readiness-for-implementation. This evolution could be informed by operational feedback from early adopters and by ongoing advances in RDF data shapes, validation, and compression techniques for geospatial data.

Looking further ahead, there is significant potential to connect coverage ontologies more deeply with sensor, processing, and workflow models. Alignment with SSN-compatible patterns and process-flow ontologies would enable end-to-end semantic descriptions spanning observation, processing, and derived products. Such integration would support more automated reasoning about data provenance, quality, and fitness-for-purpose, which is increasingly important for analytics pipelines and AI-enabled applications.

The future outlook also includes closer integration with discovery, cataloguing, and access standards. Developing explicit mappings and profiles that link ontology-based metadata with mechanisms such as OGC API – Records, STAC, and CIS would allow semantic descriptions to coexist seamlessly with existing data access infrastructures. This would help ensure that ontology-driven approaches enhance, rather than disrupt, current operational practices.

Future work should also examine how the models represented in extensions of the National Imagery Transmission Format (NITF) standard could be represented in RDF/OWL-encoded ontologies. Similarly, such work could be extended to explore representation of Motion Imagery Standards Board (MISB) metadata in RDF/OWL.

Overall, the Testbed-21 work suggests a future in which geospatial semantic architectures become a standard part of the still and motion imagery technology landscape. By continuing to combine rigorous ontological foundations with pragmatic reuse of widely deployed OGC Standards, future initiatives can progressively expand interoperability, support automation, and enable more intelligent use of coverage data across diverse domains and communities.

V VALUE PROPOSITION

The outcomes of OGC Testbed-21 deliver clear value to OGC members, sponsors, and the wider geospatial community by demonstrating how ontology-based approaches can be combined pragmatically with established standards to address real operational needs. Rather than proposing a wholesale replacement of existing practices, the work shows how semantic technologies can incrementally enhance interoperability, discovery, and understanding of coverage data. The benefits can be summarized as follows.

- For data producers and system operators, the proposed hybrid approach enables richer, machine-interpretable metadata without disrupting existing coverage models, formats, or software stacks.
- For data consumers, analysts, and downstream application developers, the value lies in improved discoverability and comprehension of imagery and other coverage products.
- From an architectural and standards perspective, the work provides OGC with a concrete, testbed-validated pattern for aligning formal ontologies with operational standards.
- For sponsors and strategic stakeholders, the Testbed-21 results highlight a pathway toward greater automation, interoperability, and reuse across the data lifecycle—from observation and processing to cataloguing and exploitation.

1

INTRODUCTION

1 INTRODUCTION

Discovery and selection of coverage data (including imagery and video) requires a usable metadata model. The metadata must be able to represent all types of coverage data, including but not limited to:

- imagery
- video
- gridded data

Metadata may be embedded in the coverage data itself, or may be provided separately, sometimes referred to as a "sidecar" file.

Designs for geospatial metadata, including for coverages, have a long history. Key standards include FGDC[1], ISO 19115 (ISO 19115-1:2014), the Coverage Implementation Schema (OGC 09-146r8), SpatioTemporal Asset Catalog (SpatioTemporal Asset Catalog, OGC 25-004), GeoDCAT (GeoDCAT-AP). These are formalized in a variety of ways, from XML Schema, through UML to JSON Schema and RDF.

Independently from this, NGA is undertaking the development of an **Imagery Domain Ontology** (IDO), an OWL (W3C REC-owl2-overview-20121211) ontology for describing imagery. The rationale for this work is to provide a formal and consistent description of imagery. IDO is to be based on the Common Core Ontologies (CCO), which in turn builds on the Basic Formal Ontology (BFO, BFO-2020). A preview of a draft version of the IDO with examples was provided to guide this project. The primary concern of the examples is geo-registration – i.e. linking image pixel indices with the geospatial ground coordinates that they represent.

Meanwhile, existing OGC Standards provide relevant capabilities that are well established in the geospatial community, and have significant software support. Data models and ontologies developed in OGC Standards are the result of a long evolution, with influences coming from many prior initiatives. Full consistency between the standards is a goal, but is always incomplete because of a need to maintain compatibility with existing deployments. Furthermore, the data models and ontologies are modular and generally self-contained, with the intention that an application can be assembled from these components. The metadata standards in particular have a pragmatic design and flat structure.

The draft version of the IDO is a strongly axiomatized OWL ontology, which is grounded in a comprehensive upper ontology (BFO) and mid-level ontologies (CCO). These enforce a high level of rigour and consistency. While this consistency is highly desirable in principle, it depends on assimilating and adopting the overall BFO ontology philosophy, which ignores the existing infrastructure and practices of the geospatial community. For example, IDO uses the CCO

[1] https://www.fgdc.gov/metadata

GeospatialOntology module, within which components for coordinates and coordinate systems duplicate capability from the OGC standards.

Given this background, the primary goal of this activity in OGC Testbed-21 was to understand the BFO/CCO ontology design approach used in the draft version of the IDO, and identify opportunities for the integration of components from the existing OGC baseline into an ontology for coverage description. The focus is on two areas:

1. description of the image geometry; and
2. description of the structure of the variables in the image content.

This report does NOT address:

- the sensor model and processing chain and how this might relate to SSN;
- a full alignment of IDO with CIS;
- alignment of IDO with STAC;
- alignment of IDO with ISO 19115; and
- alignment of IDO with DCAT.

The draft version of the IDO referred to in this report is an example ontology artifact prepared by NGA solely for OGC Testbed 21 reporting and demonstration activities. It is based on a point in time developmental ontology slim and is not an official NGA, GMSB, or IDO ontology release. The IRIs, labels, definitions, axioms, and module structure in the ontology described here should not be treated as authoritative, stable, reusable, or suitable for incorporation into OGC baseline standards, operational systems, or future testbed activities. Future work may reference officially published GMSB IDO releases when they become available.

2

SCOPE AND PRECEDENTS

2 SCOPE AND PRECEDENTS

The aims of this activity are to develop an ontology for description of coverages, in particular images and video (moving imagery).

2.1. Requirements

2.1.1. Content Requirements

GIMI metadata should ultimately support the following requirements:

- geographic extent
- temporal extent
- spatial & temporal resolution
- spatial accuracy
- acquisition parameters (including sensor model)
- processing parameters
- quality parameters
- cloud coverage
- observation dimensions or bands

For Testbed 21, the scope of the GIMI metadata is limited to the following:

1. geographic extent, georeferencing[2]
2. observation dimensions or bands

[2] https://isotc211.geolexica.org/concepts/2847/

2.1.2. Interoperability Requirements

3. GIMI metadata shall be expressed in RDF, serialized as TTL and JSON-LD (Annex C)

4. spatial elements shall use OGC GeoSPARQL geometry serialization (OGC 22-047r1)

5. other metadata elements shall be compatible with existing geospatial standards when possible

2.1.3. Sponsor requirements

6. GIMI metadata defined by an OWL Ontology (W3C REC-owl2-overview-20121211), based on CCO.

Of primary interest are the compact Turtle representation, or the RDF JSON format JSON-LD (W3C REC-json-ld11-20200716).

NOTE: The use of BFO and CCO as the basis for an D&I *Ontology* is mandated in a DoD memorandum[3].

2.1.4. Strategy

The BFO/CCO framework establishes a foundational hierarchy of classes and properties. Within this context the goal is adoption and integration of existing geospatial standards where these provide solutions to specific requirements. External components can be classified within BFO/CCO, and where the external components are based on RDF integration into IDO is mechanically straightforward. However, where the external components are not RDF, other strategies for alignment must be used.

While OWL-based metadata can be serialized using JSON, this does not mean that JSON-based data structures can be trivially blended with RDF/OWL. This report proposes solutions that integrate capabilities from OGC standards into a prototype metadata ontology based on CCO.

[3] https://dailynous.com/wp-content/uploads/2024/03/memo-dod-applied-ontology.pdf

2.2. Precedents

2.2.1. Imagery Domain Ontology (IDO)

The sponsors of the Testbed supplied a 'slim' extract from the draft version of the Imagery Domain Ontology (IDO), along with sample metadata files, to guide the work on Ontologies for Coverage Description[4]. The sample metadata files:

- list image coordinates (index pairs);
- list ground coordinates (longitude-latitude pairs);
- tie together each pair of image- and ground-coordinates in a correspondence;
- group the correspondences in a correspondence group; and
- link the correspondence group to the image.

No other concerns are mentioned in the sample data provided.

The IDO ontology extract and examples focussed on the elements required to register an image with the target region on the ground. This requires encoding geospatial coordinates in the context of an RDF/OWL instance.

The IDO extract was based on Common Core Ontologies (CCO), which in turn builds on the Basic Formal Ontology (BFO, BFO-2020). BFO and CCO are a consistent set of highly axiomatized ontologies that have been developed through an extensive, rigorous process. BFO includes some spatial concepts (e.g. spatial region, continuant fiat boundary) and the CCO GeospatialOntology module extends this with further concepts, and also adds spatial representations (coordinates, coordinate reference systems)[5].

The draft IDO supported this through a small set of classes with their associated properties:

- `Image` (from CCO)
- `TAI Timestamp`
- `Ground Coordinate`
- `Image Coordinate`
- `Correspondence`

[4]This was a point-in-time developmental ontology and not an official NGA, GMSB, or IDO ontology release.

[5]https://github.com/CommonCoreOntology/CommonCoreOntologies/blob/develop/src/cco-modules/GeospatialOntology.ttl

- `Correspondence Group`

These are all subclassed from Information Content Entity branch of the BFO/CCO Ontologies — i.e., they are concerned purely with the information content, and not the things in the real world.

Key classes in the ontology are shown in this class hierarchy, which is composed of BFO classes at the top level, CCO classes from the GeospatialOntology[5] and InformationEntityOntology[5] at the mid-level, and specialised subclasses for imagery metadata at the base:

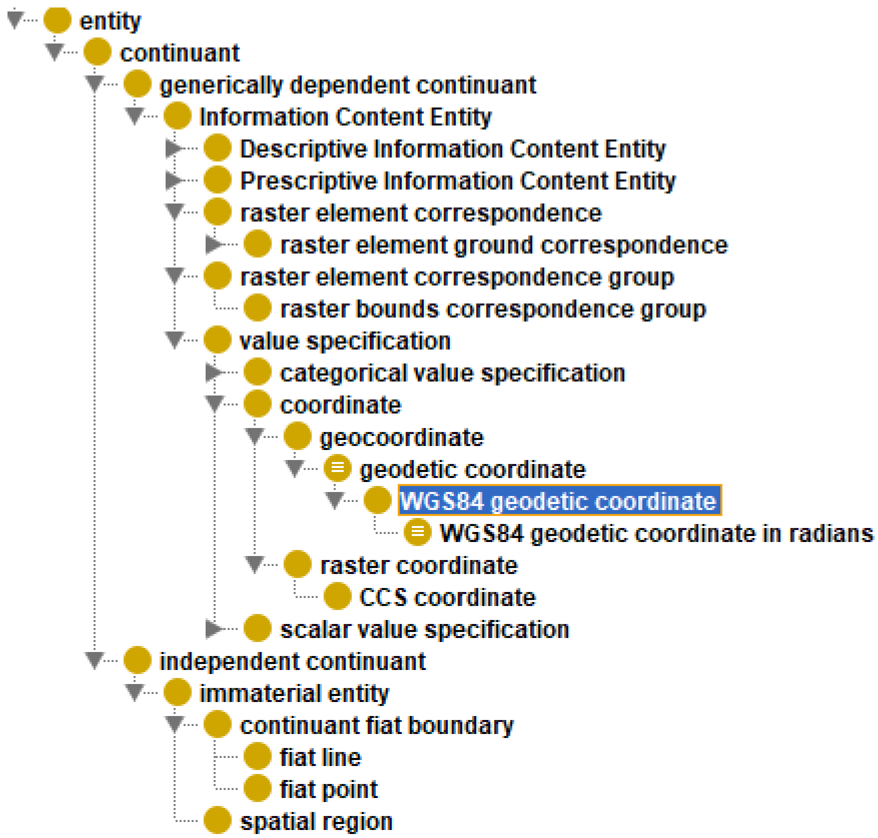

Figure 1 — Slim IDO class hierarchy

2.2.2. OGC baseline

A number of OGC Standards are relevant to the requirements. In particular:

- GeoSPARQL is a moderately axiomatized RDF/OWL ontology for representing geospatial features and geometries (OGC 22-047r1);
- GeoJSON is a JSON specification for representing geospatial features and geometries (IETF RFC 7946) (GeoSPARQL adopts GeoJSON as a geometry serialization option);
- Coverage Implementation Schema (CIS) is a model for coverage data, including imagery, with a JSON-Schema implementation and JSON-LD context (OGC 09-146r8, JSON-Schema for CIS, JSON-LD context for CIS);

- SWE Common is a schema language for encoding complex variable structures, with a JSON-Schema implementation (OGC 24-014);
- Semantic Sensor Network Ontology (SSN) is a weakly axiomatized OWL ontology for describing data capture by sensors, as well as sampling and actuation (W3C REC-vocab-ssn-20171019, W3C WD-vocab-ssn-2023-20250916);
- STAC is a JSON specification (building on GeoJSON) for describing geospatial assets, with a catalog API (SpatioTemporal Asset Catalog, OGC 25-004); and
- GeoDCAT is a profile of DCAT that adds elements for geographic information, and defines specific patterns compatible with the expectations of the geospatial community, paticularly as embodied in ISO 19115-1:2014 (W3C REC-vocab-dcat-3-20240822, GeoDCAT-AP).

NOTE:Where flaws are found in the components published by OGC or other organizations, corrections to these should be fed back to the component maintainers, rather than re-implementing the functionality in a new schema or ontology.

2.2.2.1. GeoSPARQL

GeoSPARQL (OGC 22-047r1) is the OGC standard for representing geospatial data in RDF. GeoSPARQL provides an RDF implementation of the basic concepts *Feature* and *Geometry* from the GIS community.

For Geometry, GeoSPARQL reuses classes and encodings from Simple Features (OGC 06-103r4, [ISO19125]) and CRS references to the EPSG registry (EPSG-API). Geometries are encoded in formatted literals in the following formats: GML, WKT, GeoJSON, KML, and DGGS (GeoSPARQL Section=10.8). For example, these basic WKT and GeoJSON literal encodings describe same point:

```
"POINT(145.112566 -37.848910)"^^geosparql:wktLiteral

"""{
  "type": "Point",
  "coordinates": [145.112566, -37.848910]
}"""^^geo:geoJSONLiteral
```

Listing 1 — GeoSPARQL literals

Those examples use the default CRS, which is WGS84 lon-lat, decimal degrees. However, WKT also allows for the indication of a *Spatial Reference System* (SRS) so WKT geometry may be given using any Coordinate Reference System (CRS). For example, these two literals describe the same location, in WGS84 lat-lon decimal degrees, and in UTM zone 55H metres:

```
"""
<http://www.opengis.net/def/crs/EPSG/0/4326> POINT(-37.848910 145.112566)
"""^^geosparql:wktLiteral

"""
<http://www.opengis.net/def/crs/EPSG/0/32755> POINT(333941.44 5809269.94)
"""^^geosparql:wktLiteral
```

Listing 2 — WKT CRS support

The order of the axes is determined by their arrangement in the definition of the referenced CRS. Further background on GeoSPARQL is given in Annex D.

2.2.2.2. CIS

OGC (and the ISO Geographic Information committee TC 211) have produced a system of standards for “coverages” — i.e., data that vary over space and/or time, such as satellite imagery, climate data, or elevation grids. The Coverage Implementation Schema (CIS) is the primary implementation for coverage data (OGC 09-146r8).

CIS is very general within which regular gridded coverages are the most common special-case, with 2D imagery being a special case of those. 3D (2D+elevation, 2D+time) and 4D domains are other possibilities. CIS is based on the “domain-range” structure. The coverage *domain* is the set of locations for which values are provided in the coverage *range*. The coverage domain may be specified in various ways, including bounds and resolution per axis, or explicitly enumerated position sets, and various representations for transformations.

NOTE: The term “domain” is used in the CIS to refer to the set of locations that the coverage relates to — domain <coverage>. This is different to “domain” in the context of the “Imagery Domain Ontology”, where it refers to the area of human knowledge concerning imagery — domain <general vocabulary>

The coverage range is specified in two parts:

- `rangeType`, the definition and schema or structure for the variables in the coverage; and
- `rangeSet`, which are the actual values of the variables.

The **rangeType** definition is particularly important for any coverage (including imagery) where the values are not simple scalar values. It is necessary to understand the elements and arrangement of the range values, including vectors, and more complex structures.

Within CIS, the schema for the rangeType is provided by OGC SWE Common (OGC 24-014). Examples of CIS in RDF are provided:

- in two RDF syntaxes (JSON-LD and N3) covering the full scope of CIS ; and
- in JSON-LD for data on purely regular grids.

Further background on CIS is given in Annex D.

2.2.2.3. SWE Common

OGC SWE Common is a data model for complex data (OGC 24-014). It allows for the definition of the record structure for scalars, vectors, matrices, and streams, with simple or complex values at any point, built up from the usual numeric, textual, categorical and Boolean types, and geometries and times.

SWE Common is specified as a UML model, with an implementation of the record structure (schema) in JSON, with data values in JSON, text, or binary. Geometries are encoded using the JSON array-of-arrays syntax for sets of coordinates that build up higher-order geometries.

A canonical JSON Schema is available for SWE Common (JSON-Schema for SWE Common). The structure of the SWE DataRecord documented in [SWE_Common_v3] provides a useful starting point for describing the range-type of a coverage. In particular, see SWE Record Components JSON Schema (with short examples) and SWE DataRecord JSON Encoding Rules Examples.

There is currently no OWL implementation of the SWE Common Record Object. JSON-LD contexts for SWE Common are provided for CIS (JSON-LD context for CIS). This includes a context for rangeType[6] which is primarily concerned with SWE Common terms. However, this is incomplete as an RDF implementation, as discussed in this long issue thread in GitHub and there are inconsistencies between the JSON-LD and JSON Schema[7].

It should be noted that SWE Common is closely aligned with the Sensor Model Language (SensorML), which is also from the Sensor Web Enablement (SWE) suite of standards. Both of these standards now play a prominent role in the OGC API suite of standards through their use in the OGC API – Connected Systems standard.

Further background on SWE Common is given in Annex D.

2.2.3. Geospatial Metadata and catalog standards

OGC has also published or collaborated in the development of a number of standards for describing metadata and catalog interfaces using them. These include ISO 19115-1:2014, OGC 25-004, OGC 20-004r1, GeoDCAT-AP.

The OGC metadata standards have not been used in this project.

Further background on these metadata standards are given in Annex D.

[6] https://github.com/opengeospatial/coverage-implementation-schema/blob/main/standard/schemas/1.1/rdf/rangetype-context.json

[7] https://github.com/opengeospatial/coverage-implementation-schema/issues/26

3

PROPOSED ONTOLOGY FOR COVERAGE DESCRIPTION

3 PROPOSED ONTOLOGY FOR COVERAGE DESCRIPTION

This section describes modules for an ontology used for metadata files covering the scope of the sample data provided for Testbed 21. These satisfy the requirements for this project as identified in the scope section, specifically:

- build on CCO following the patterns provided in the IDO snapshot as much as possible;
- use GeoSPARQL (OGC 22-047r1) for spatial information, and clarify the geospatial semantics using an explicit real-world referent for imagery; and
- describe coverage cell-properties using SWE Common JSON encoding (at least in the short term).

The proposed ontology serves as a recommendation from Testbed-21 for future versions of the IDO. Modules implementing these requirements are described in the following sections.

3.1. Geometry representation using GeoSPARQL

The OGC Standard for spatial data in RDF/OWL is GeoSPARQL (OGC 22-047r1). This uses geometry encodings from Simple Features (OGC 06-103r4) and coordinate definitions from ISO 19111 (ISO 19111:2019). GeoSPARQL is supported by several standard RDF implementations, triple-stores and graph databases[8].

On the other hand, IDO sits in the BFO/CCO framework (BFO, BFO-2020, CCO), within which the CCO GeospatialOntology module duplicates some of the capability from the OGC Standards. Design decisions in the CCO GeospatialOntology introduce limitations in two areas:

1. Individual coordinates within a single position are on separate nodes in the RDF graph[9];
2. The coordinate reference system (CRS) formalization is not based on existing and historical standards.

[8] Oracle Spatial, AllegroGraph, Neo4j Spatial, GraphDB GeoSPARQL Plugin support GeoSPARQL directly. A GeoSPARQL Compliance Benchmark evaluates GeoSPARQL compliance of 9 RDF triplestores.

[9] Clause 4.1 includes a discussion of why coordinates are not independent and should not be on separate graph nodes.

Immediate consequences of these choices are that: (i) spatial data processing cannot directly use geometry function support in existing software platforms, and (ii) the CRS is not denoted by an identifier from the existing comprehensive standard CRS catalogues which are used ubiquitously by standard software (EPSG Geodetic Parameter Dataset).

To enable IDO metadata to utilise geospatial standards consistently, we define a formal alignment between GeoSPARQL and BFO/CCO which:

- allows position and geometry to be encoded using GeoSPARQL datatypes (`WKTLiteral`, `GeoJSONLiteral` etc.);
- provides access to geometry-types from Simple Features (`Point`, `Polygon`, `Surface`, `Envelope`, etc.); and
- supports use of CRS definitions either (i) accessed by a IRI reference to a standard catalogue[10], or (ii) in WKT definitions inline (OGC 18-010r11).

The alignment may be summarized as follows:

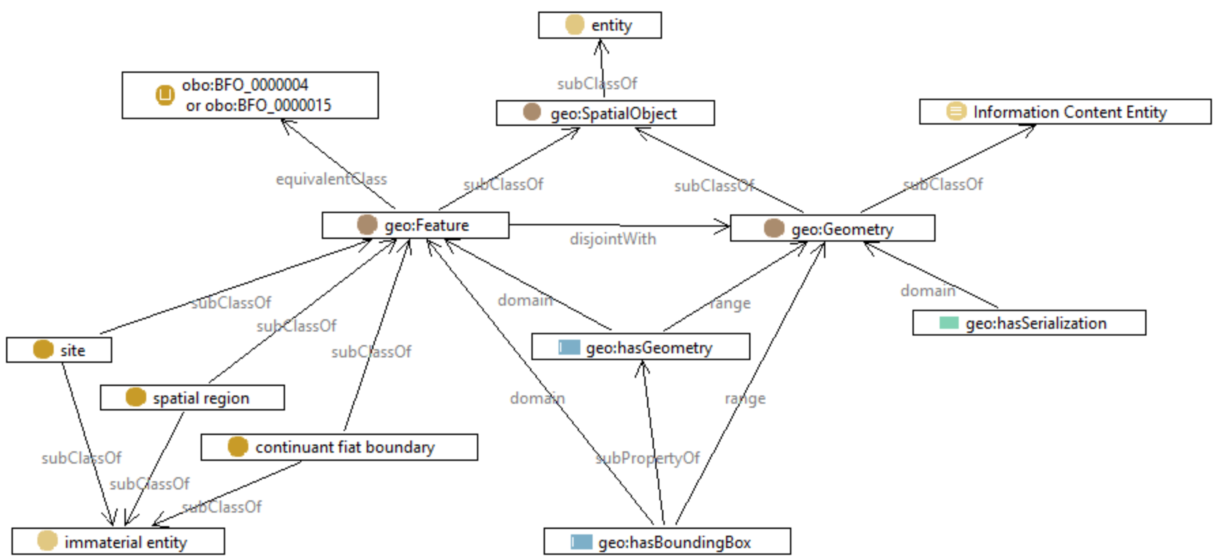

Figure 2 — GeoSPARQL-BFO alignment summary

The alignment axioms are tabulated in Annex B and packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/rdf/ontology/tb21-geosparql.ttl

[10]EPSG REST API https://apps.epsg.org/api/swagger/index.html , e.g. https://apps.epsg.org/api/v1/CoordRefSystem/4326

The key to the alignment is that the BFO "immaterial entity" subclasses[11] are kinds of GeoSPARQL Feature. The geometry of a GeoSPARQL Feature is described by a GeoSPARQL Geometry[12] that is serialized as a WKT literal[13]. Thus, the position and shape of any individual in the BFO "immaterial entity" subclasses can be expressed in WKT.

NOTE 1: The strategy of using a formatted literal to embed an external encoding in RDF is an example of a particular style of *encapsulation*. See Clause 4.2 for more details.

NOTE 2: A non-normative GeoSPARQL-BFO alignment was provided in the GeoSPARQL documentation (OGC 22-047r1). This has been discovered to have some flaws. The alignment presented here has been submitted to OGC as a change-request for GeoSPARQL.

3.2. Coverage domain — region of interest

The sample data provided by the Sponsor for review in the testbed packaged information about imagery (corner indexes, time-stamp) along with some geospatial positions, and then asserts correspondences between the image corners and positions. However, the semantics of the geospatial positions included in the file is not explicit. Since the ontology for coverage description is meant to fit into a rigorous semantic context, this was deemed unsatisfactory.

The imagery that we are concerned with intentionally represents some part of the real world. The referent has geospatial location and shape. In OGC terms, this is the *domain* of the imagery (domain <coverage>), and is the information required to associate imagery or other observations to the real world. To support the proper semantics, we introduce a class `"region of interest"`.

"Region of interest" is defined as "Spatial region that is the subject of investigation or observation". It is a subclass of "spatial region" and thus of `geo:Feature`. It may be the subject of imagery, or of interpretation.

Region of interest has any dimensionality (from "`spatial region`"). We add the following geometric properties:

- "`has boundary`" *some* "`continuant fiat boundary`"
- "`has bounding box`" *some* `sf:Envelope` (from `geo:Feature`)
- "`has geometry`" *some* `sf:Geometry` (from `geo:Feature`)

NOTE:The *bounding box* (Envelope) is a rectangle specified by two points – the minimum (lower left) and maximum (upper right). This is not generally the same as the *boundary*, which (a) might not

[11]"spatial region", "site", and "continuant fiat boundary" and the latter's subclasses "fiat point", "fiat line", "fiat surface" — see https://github.com/bfo-ontology/bfo-2020#basic-formal-ontology-hierarchy

[12]includes *Simple Features* Geometry types

[13]or alternatively GeoJSON, KML, GML or DGGS

be rectangular, or (b) might not be oriented in alignment with the axes of the bounding box (see OGC 06-103r4).

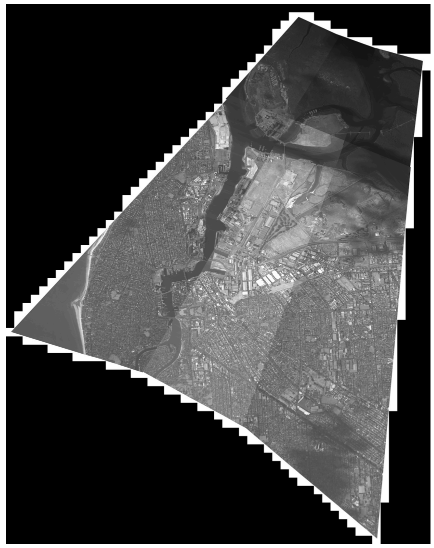

Figure 3 — Irregular region where the bounding box is not the same as the boundary

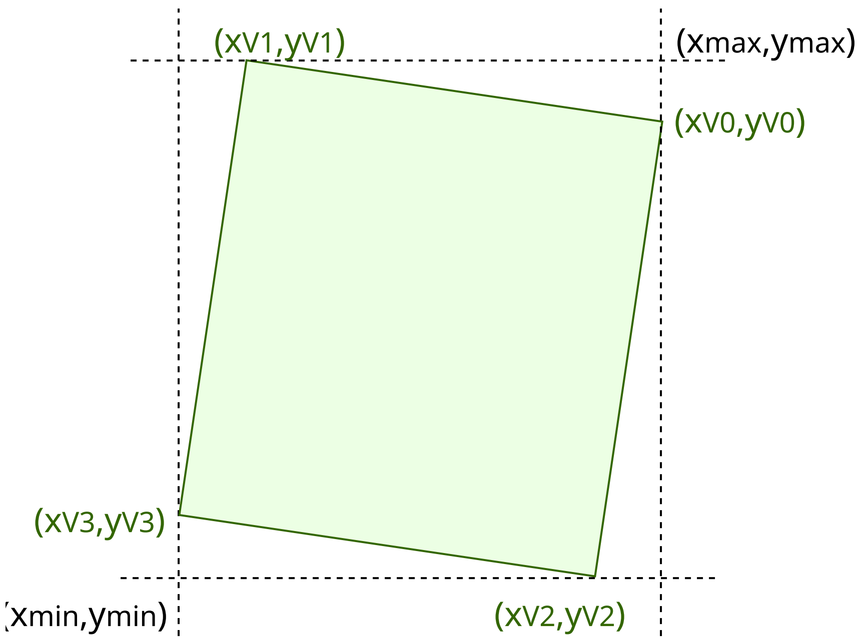

Figure 4 — Rotated region where the bounding box is not the same as the boundary

Vertex points are associated with the *actual* boundary. These points are separate nodes in the graph, as they must carry independent identifiers in order for them to be part of correspondences (see next section).

The definition of "region of interest" may be summarized as follows:

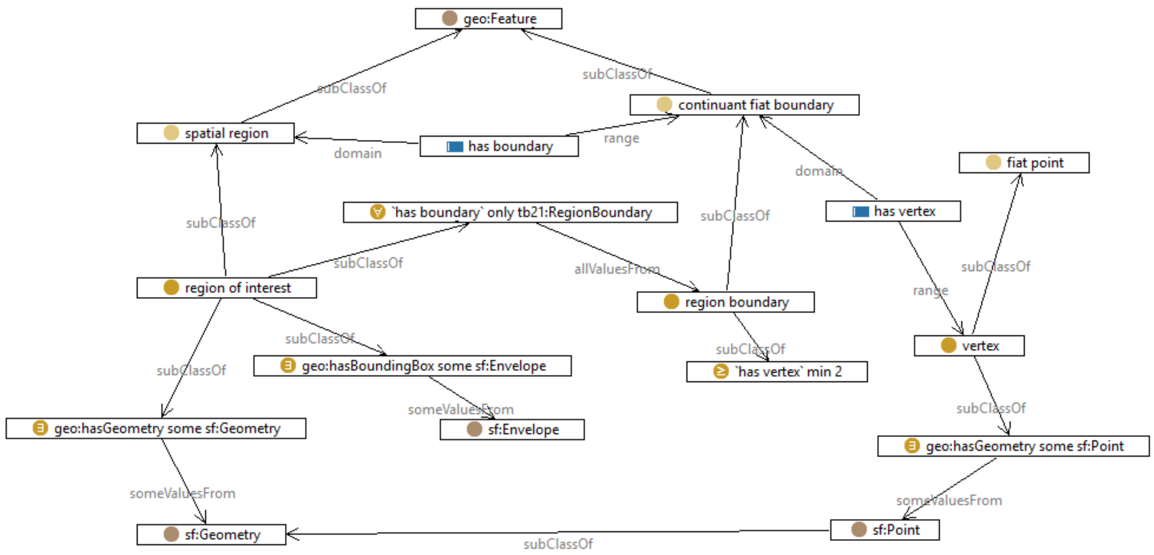

Figure 5 — Region of interest summary

The “region of interest” axioms are tabulated in Annex B and packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/rdf/ontology/tb21-region.ttl

3.3. Image properties

Imagery and other coverage data can describe many properties of a region of interest. The OGC Standard for describing ‘coverages’ is CIS (OGC 09-146r8), within which the following blocks of data are specified:

- `Envelope` and `DomainSet` – the image extent and its discretization;
- `RangeType` – describes the semantics and structure of cell or pixel values; and
- `RangeSet` – gives the actual values for the positions or cells in the domain.

The Domain and the RangeType are key elements of the coverage *metadata*.

3.3.1. Image class

For TB21 we introduce an “Image (of a region)” class that is a subclass of CCO “Representational Information Content Entity”. The range of “represents” is restricted to only “region of interest”.

3.3.2. Image geometry

The corner coordinates of the image are expressed as pixel-indexes. These are mapped to ground coordinates related to the image domain (“region of interest”) in image correspondences.

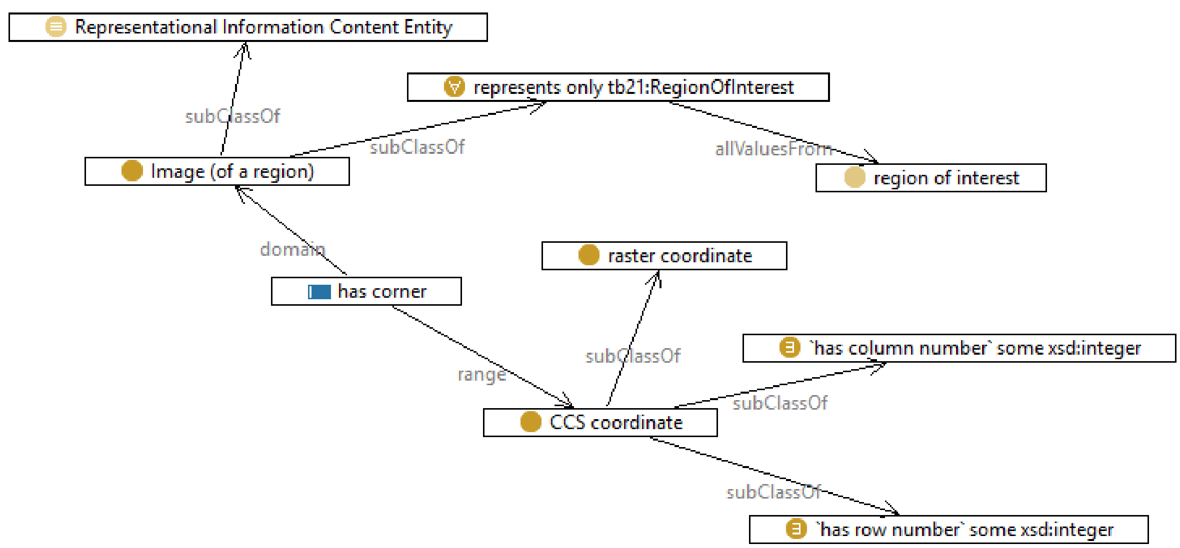

Figure 6 — Image corner coordinates

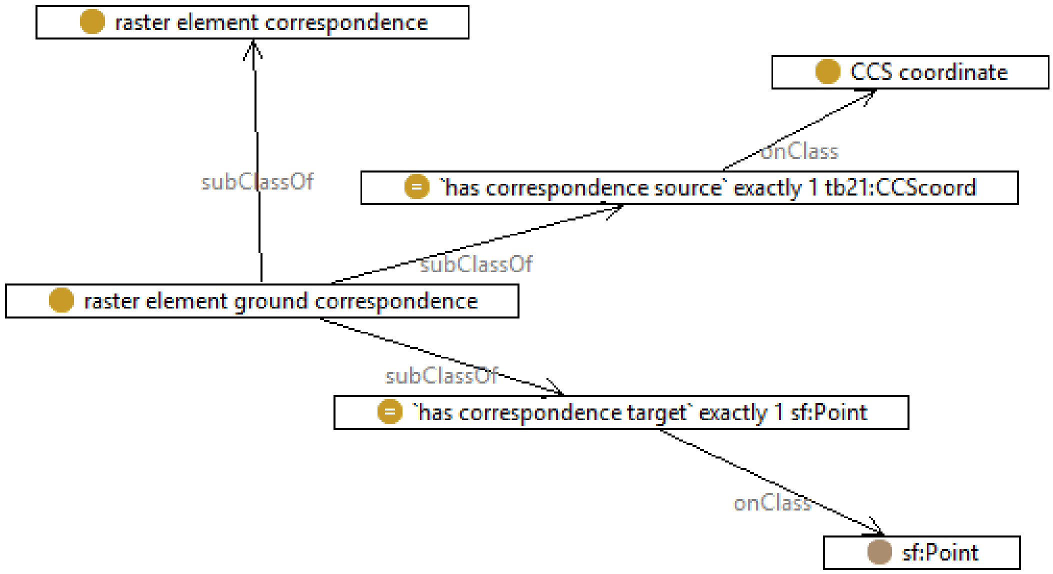

Figure 7 — Image correspondence structure

3.3.3. Cell properties

For the RangeType CIS uses the `DataRecord` encoding from SWE Common (OGC 24-014). Use of common components in the ontology for coverage description will make interoperation easier. The Testbed-21 participants thus propose the use of CIS RangeType in the GIMI metadata.

The normative implementation of SWE Common is in JSON, defined in version 3 of SWE Common, for which a JSON Schema is available[14]. While this is not an OWL ontology, JSON data conformant to SWE Common may be included in RDF as a literal using the `rdf:JSON` datatype. This is the recommended approach at least in the short term. In the context of CCO-based systems, the "RangeType" with a SWE DataRecord is a kind of "Prescriptive Information Content Entity".

NOTE: The strategy of using a formatted literal to embed an external encoding in RDF is an example of a particular style of *encapsulation*. See Clause 4.2 for more details.

[14] https://schemas.opengis.net/sweCommon/3.0/json/

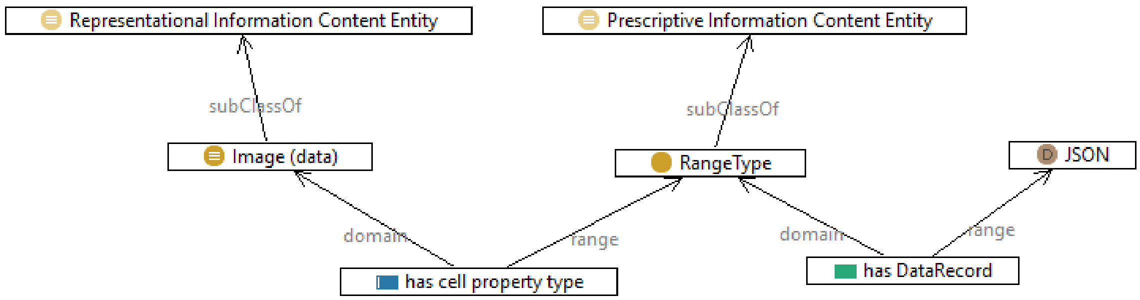

Figure 8 — Image cell properties summary

Would it be possible to define a SWE Common JSON-LD matching the normative JSON? CIS provides some JSON-LD contexts for parts of SWE Common[15]. However, a JSON-LD context only maps JSON keys to IRIs, and there is no corresponding OWL Ontology for SWE Common.

Furthermore, basic incompatibilities between JSON and RDF mean that even if an OWL ontology implementation for SWE Common was developed, its JSON-LD serialization would not reliably match the SWE Common JSON. In particular: JSON supports ordered arrays but not repeated keys, while RDF allows repeated properties (keys), but array implementation is a kludge. Workarounds have been proposed, but require additions to appear inside the data that are not supported by standard RDF or JSON libraries.

3.3.4. Overall image properties

The overall image properties are summarized as follows:

[15] https://schemas.opengis.net/cis/1.1/rdf/ — though there are mismatches with the terms in the normative JSON implementation.

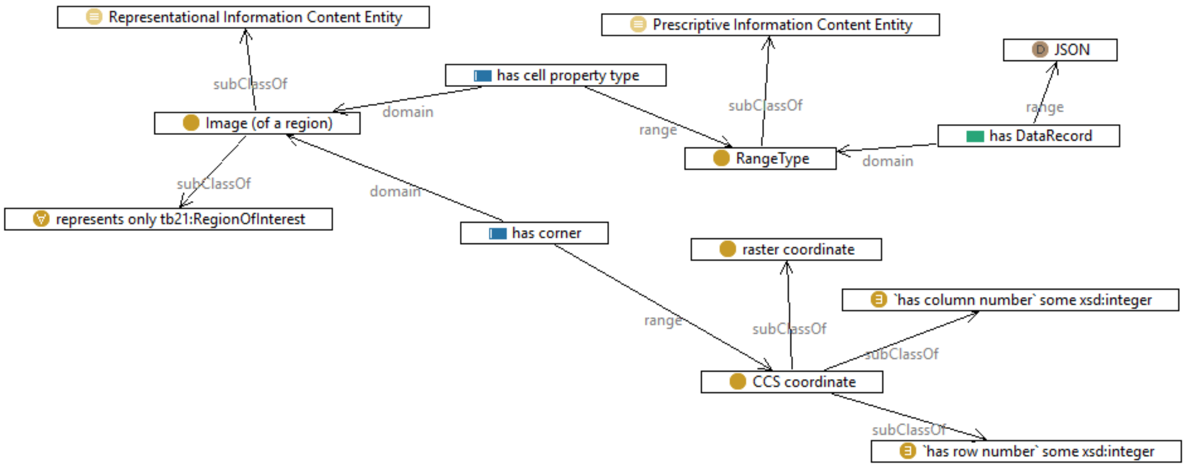

Figure 9 — Image summary

The image axioms are tabulated in Annex B and packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/rdf/ontology/tb21-image.ttl

3.4. Complete example

The design and axiomatization described over the previous sections involves some deep subsumption hierarchies, and sometimes requires use of object properties specified on super-classes. Pulling this all together is best illustrated using a worked example.

A provisional example for a single image based on IDO was provided by the sponsors. This has been redesigned using the modifications discussed above (Clause 3.1, Clause 3.2, Clause 3.3).

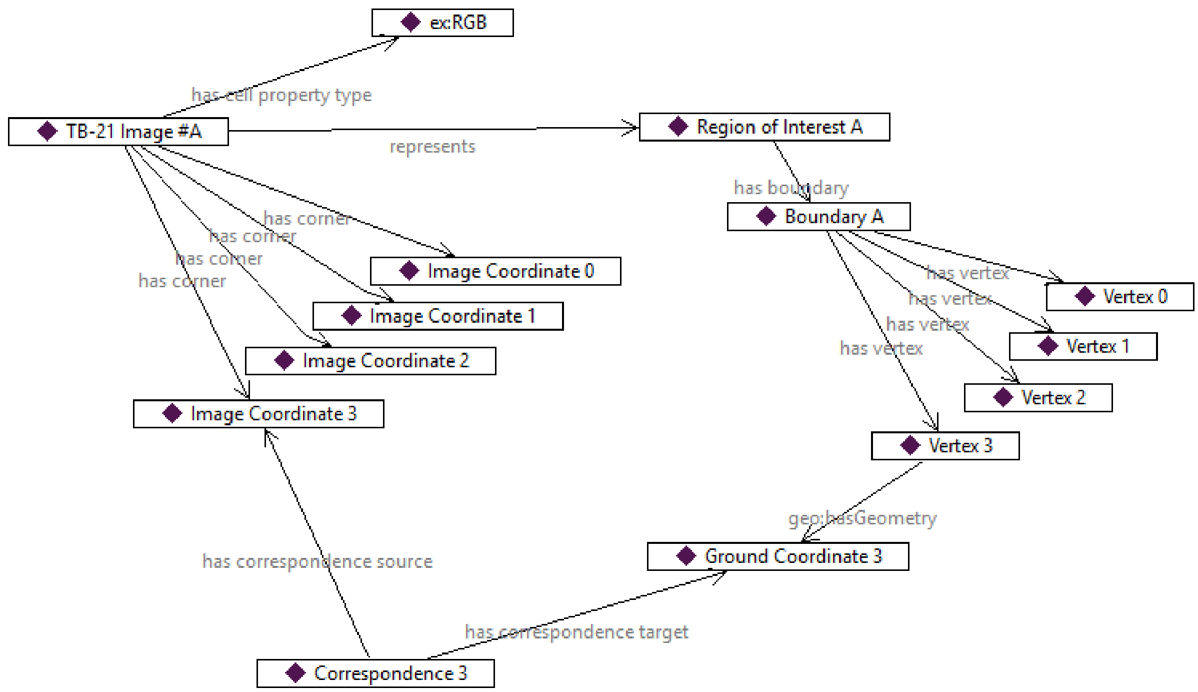

Figure 10 — Complete example based on sample data provided by sponsors

Two of the nodes shown have Literal values.

The RangeType ex:RGB has a DataRecord serialized in JSON:

```
ex:RGB
    a tb21:RangeType ;
    tb21:hasDataRecord """
    {
    "name": "RGB Image",
    "type": "DataRecord",
    "fields": [
        {
        "label": "Red",
        "description": "Red - 510-750 nm",
        "type": "Count"
        },
        {
        "label": "Green",
        "description": "Green - 440-600 nm",
        "type": "Count"
        },
        {
        "label": "Blue",
        "description": "Blue - 415-564 nm",
        "type": "Count"
        }
    ]
    }
      """^^rdf:JSON ;
.
```

Listing 3

The Geometry labelled “Ground Coordinate 3” is serialized as a `geo:WKTLiteral`

```
<urn:uuid:c44e4b92-7317-411f-a10f-3cc1007aee90>
  a sf:Point ;
  rdfs:label "Ground Coordinate 3" ;
  geo:hasSerialization """
  <http://www.opengis.net/def/crs/OGC/1.3/CRS84> POINT(138.494192 -34.804844)
  """^^geo:wktLiteral ;
.
```

Listing 4

Other geometries are serialised in the same way.

This example is packaged in this ontology file

https://github.com/opengeospatial/coverage-description-ontology/rdf/example/tb21-single-image-hevc.ttl

4

ADDITIONAL CONCERNS

4 ADDITIONAL CONCERNS

This section provides further discussion of some specific concerns.

4.1. Coordinate representation in OWL/RDF graphs

Should individual coordinates be captured in individual nodes?

RDF has a directed graph model. Each item of information is a node in the graph. Links are based on semantics. This is highly effective for reasoning and logic.

It is tempting to stay in the graph paradigm for coordinates, and put each element of a geometry representation on a node. This was the approach taken in the CCO GeospatialOntology and the prototype IDO.

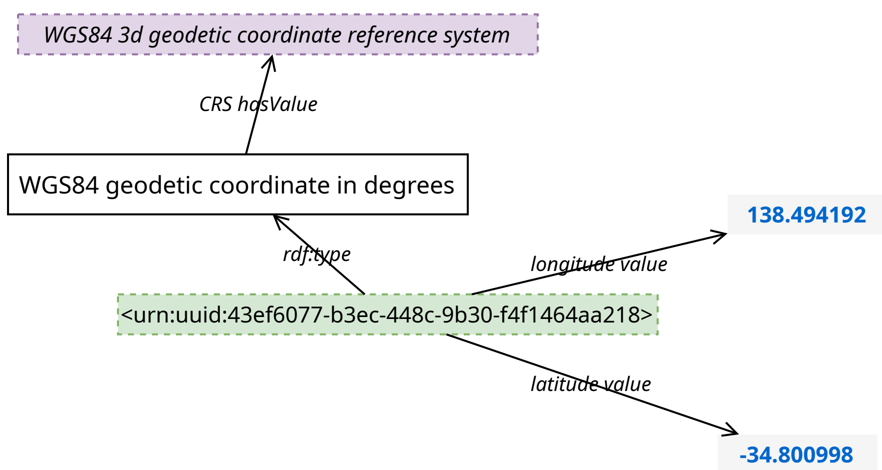

Figure 11 — Coordinates expressed as individual nodes in a graph

However, this has significant drawbacks, because of the following.

1. Coordinates in use may refer to different axes, including east, north, south, west, left, right, up, down, elevation, altitude, and others. So a native RDF representation would require properties for all these axis sets, alongside latitude & longitude.

2. Coordinate values may be referenced to different origins and datums, such that exactly the same *position* is denoted by very different *numbers* when using different coordinate reference systems. Each CRS must be precisely described.

3. For a higher order geometry (polygon, surface, solid, etc) it is necessary to link individual positions together in a sequence, or with particular topology.

4. Coordinate values for a position are not independent of each other, and should not be inspected, transported or manipulated separately; Geometry computations operate on positions[16], not individual coordinates[17].

5. It is inconsistent with industry standards.

Accommodating the first three of these requirements in an ontology would result in a combinatorially large number of very similar classes and properties, and a whole theory of geometry. What should be a basic representational challenge would become a significant scalability issue.

Meanwhile, standards exist for the representation of position, and for geometries built up from multiple positions, primarily based on the ISO/OGC Simple Features standard (OGC 06-103r4, ISO 19125-1:2004). Implementations in WKT and GeoJSON are the most commonly used. These formats (i.e. datatypes) manage the arrays and orderings necessary.

WKT also allows coordinates to be annotated with the CRS used, which is usually denoted by a IRI (but may be included inline).

```
"POINT(145.112566 -37.848910)"^^geosparql:wktLiteral

"""
<http://www.opengis.net/def/crs/EPSG/0/4326> POINT(-37.848910 145.112566)
"""^^geosparql:wktLiteral

"""
<http://www.opengis.net/def/crs/EPSG/0/32755> POINT(333941.44 5809269.94)
"""^^geosparql:wktLiteral
```

Listing 5 — These WKT literals all describe the same point

A CRS IRI can dereference to a CRS definition, usually taken from a standard CRS registry such as that maintained by EPSG (EPSG Geodetic Parameter Dataset). The registry maintained by EPSG has more than 8000 CRS, using nearly 300 distinct axes, and hundreds of other elements like datums, units of measure, etc. Each of these is denoted by an IRI (EPSG-API).

Reproducing these capabilities in an Ontology is an unnecessary extravagance.

4.1.1. Semantics vs. Mathematics

GeoSPARQL (OGC 22-047r1) adopts these datatypes for geometry representation. Conceptually, this respects the distinction between semantics/logic and mathematics (including geometry). Mathematical operations are not the same as logic operations. You can *reason* with logic, but *calculate* with numbers. *Semantic relationships* are expressed using RDF predicates. *Mathematical entities*, including their relationships, are encoded in literals with well defined datatypes.

[16] https://isotc211.geolexica.org/concepts/347/

[17] https://isotc211.geolexica.org/concepts/2038/

Geometric operations like “does this point lie within that polygon?” are of a different kind to logic operations like “is this individual a member of that class?”. Elements of the geometry representation (e.g., coordinates) should not be in separate nodes in a semantic graph because **they are not semantically independent**.

A geospatial ontology must implement geospatial concepts. These might be *attached onto* concepts in a foundational ontology, but it won’t be competent to manage the detail unless the foundational ontology is about mathematics.

Generic OWL and SPARQL can store and pass through literals encoded with arbitrary RDFS datatypes, but do not support spatial operations. However, many graph-stores and RDF systems support spatial queries and operations defined in GeoSPARQL[8].

4.2. Encapsulation

The idealized solution for coverage metadata might address all needs in a single framework and implementation platform. However, given the variety of specific requirements, development of a single solution is onerous, will be cumbersome to maintain, and furthermore requires a matching software system that can simultaneously handle all components. It is challenging, and therefore unlikely, for a general framework to be able to reflect the state of the art across all aspects.

A primary requirement from the sponsors of this project is that coverage metadata must be formalized using an ontology based on CCO – the Imagery Domain Ontology (IDO). Meanwhile, various of the requirements are addressed by existing standards, which represent the current state of a development process by a community tackling that part of the problem. So within the IDO context, we must consider ways to integrate and utilize these components.

The BFO/CCO framework behind IDO establishes a foundational hierarchy of classes and properties (BFO, BFO-2020, CCO, see the BFO hierarchy). External components used in conjunction with BFO/CCO must be classified within this in order to be consistently used an application ontology.

However, BFO and CCO do not employ datatypes beyond the basic types built in to OWL2 — decimal, integer, boolean, dateTime, etc. Since vectors are essential for geometry (and other areas of mathematics and physics) a strategy for vector and array representation is necessary.

Where the external components are based on RDF, integration into IDO is relatively straightforward. However, where the external components are not RDF, *encapsulation* may be used.

Some datatypes are provided by the foundation RDF specifications, in particular `rdf:XML` and `rdf:JSON` (W3C CR-rdf12-concepts-20260407). This are effectively extension points in RDF.

Additional RDFS Datatypes may be designed to meet further specific requirements. These allow the encapsulation of content with embedded structure and semantics. This is particularly useful for mathematical data, including geospatial.

4.2.1. SWE Common encapsulation

A schema for range-types in CIS is provided by SWE Common (OGC 24-014). This provides a compact encoding of the structure of values in the coverage range-set. Some examples of range-types are provided in the GitLab repository.

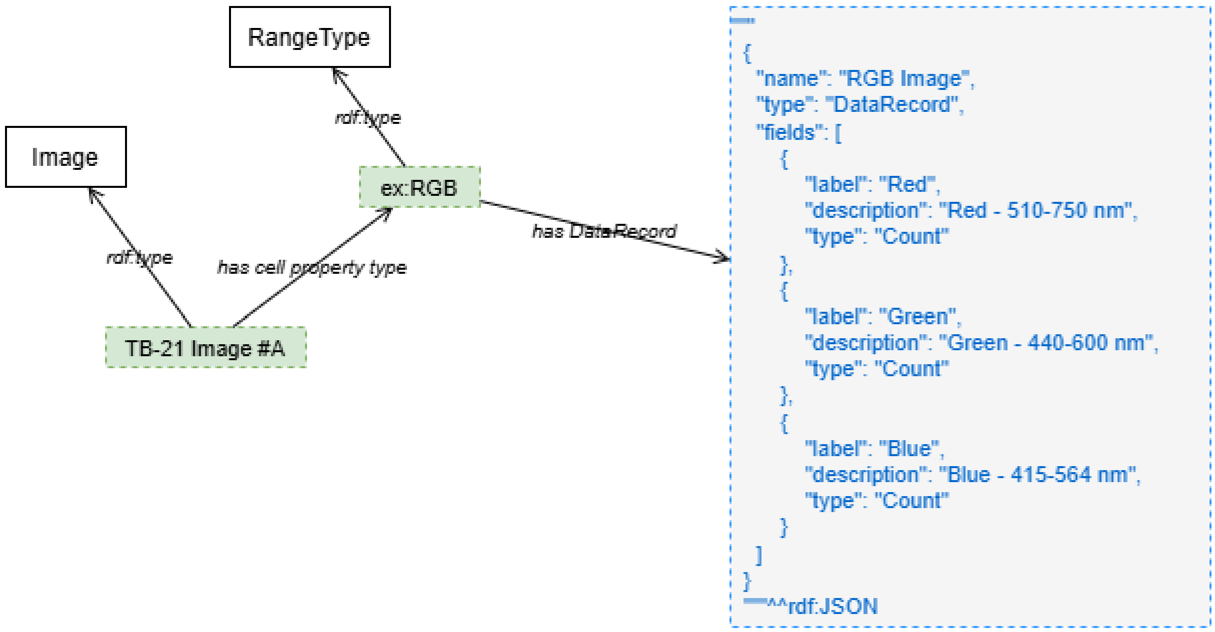

Figure 12 — SWE Common DataRecord embedded in a JSON literal within a graph

4.2.2. GeoSPARQL encapsulation

A set of coordinates and the CRS describing a position comprise a discrete information item as a whole (see explanation and examples in Annex D). The key datatype for spatial data in GeoSPARQL is `geosparql:wktLiteral`.

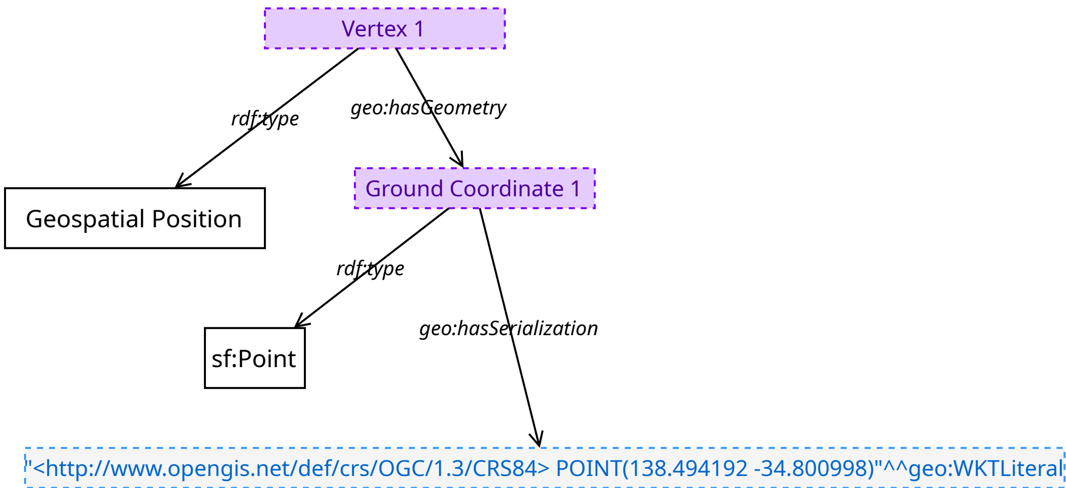

Figure 13 — Coordinates embedded in a WKT literal within a graph

In order to directly access and operate on data embedded in literals, the generic OWL/RDF processor may be supplemented to support that encoding. GeoSPARQL support is available in a number of standard products[8].

Note that individual values may be extracted from a formatted literal using standard SPARQL functions. For example, the following listing is an example of a bounding box in GeoSPARQL.

```
ex:enva
  a sf:Envelope ;
  sf:maximum ex:bbmax ;
  sf:minimum ex:bbmin ;
  rdfs:label "Bounding Box A" ;
.
ex:bbmax
  a sf:Point ;
  geo:hasSerialization """
  <http://www.opengis.net/def/crs/OGC/1.3/CRS84> POINT(138.498668 -
34.800998)"""^^geo:wktLiteral ;
.
ex:bbmin
  a sf:Point ;
  geo:hasSerialization """
  <http://www.opengis.net/def/crs/OGC/1.3/CRS84> POINT(138.494192 -
34.804844)"""^^geo:wktLiteral ;
.
```

Listing 6 — Bounding box represented as a sf:Envelope with limits encoded as WKT literals

The following query extracts the minimum and maximum values as scalars:

```
SELECT $xmin $ymin $xmax $ymax
WHERE
{
 ?env a sf:Envelope ;
  sf:minimum/geo:hasSerialization ?wktmin ;
  sf:maximum/geo:hasSerialization ?wktmax .

    BIND( strafter(str(?wktmin) , "POINT(") as ?mincp )
```

```
    BIND( strbefore(?mincp, ")") AS ?minc )
    BIND(
      xsd:decimal(
        REPLACE(
         ?minc,
          "([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d+)?) ([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d
+)?)",
          "$1"
        )
      ) AS ?xmin
    )
    BIND(
      xsd:decimal(
        REPLACE(
         ?minc,
          "([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d+)?) ([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d
+)?)",
          "$2"
        )
      ) AS ?ymin
    )

    BIND( strafter(str(?wktmax) , "POINT(") as ?maxcp )
    BIND( strbefore(?maxcp, ")") AS ?maxc )
    BIND(
      xsd:decimal(
        REPLACE(
         ?maxc,
          "([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d+)?) ([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d
+)?)",
          "$1"
        )
      ) AS ?xmax
    )
    BIND(
      xsd:decimal(
        REPLACE(
         ?maxc,
          "([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d+)?) ([+-]?(?:\\d+(?:\\.\\d*)?|\\.\\d
+)?)",
          "$2"
        )
      ) AS ?ymax
    )
}
```

Listing 7 — SPARQL query to extract the minimum and maximum values from an `sf:Envelope` encoded in a WKT literal

4.3. Sensors and sensing chain

Complete imagery description must allow for the description of the data capture chain from sensing to information, for all the relevant scenarios including but not limited to:

- Basic earth imaging;

- Hyperspectral earth imaging; and
- Imaging non-earth-based objects.

Digital imagery is the result of multiple processes, starting with the generation of energy from the subject, transmitted through various media, detected in a sensing frustum, with multiple data processing steps ending with an information artefact. Various transduction, sensing, and processing stages are involved.

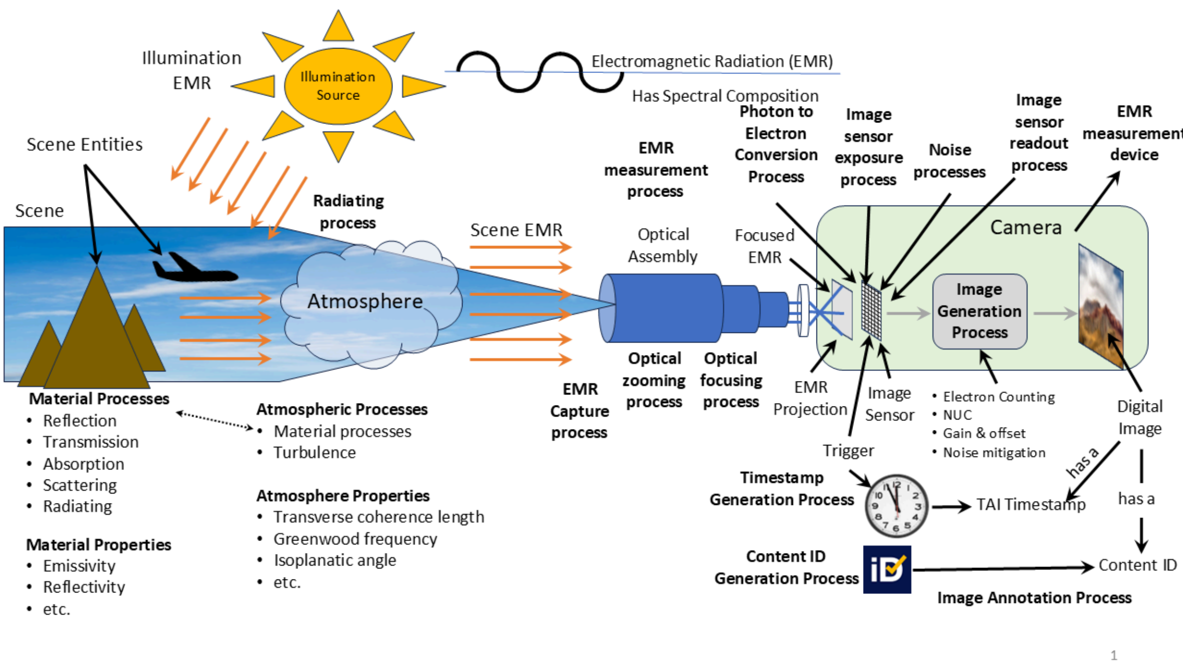

Figure 14 — General use case for imagery data capture

The process-flow model is a standard approach to describing the flow of information in a system. The W3C Provenance Ontology (W3C NOTE-prov-overview-20130430, W3C REC-prov-o-20130430) is a particularly simple implementation, which describes the result of a chain in terms of a sequence of alternating activities and entities resulting from them, providing the input to the next activity.

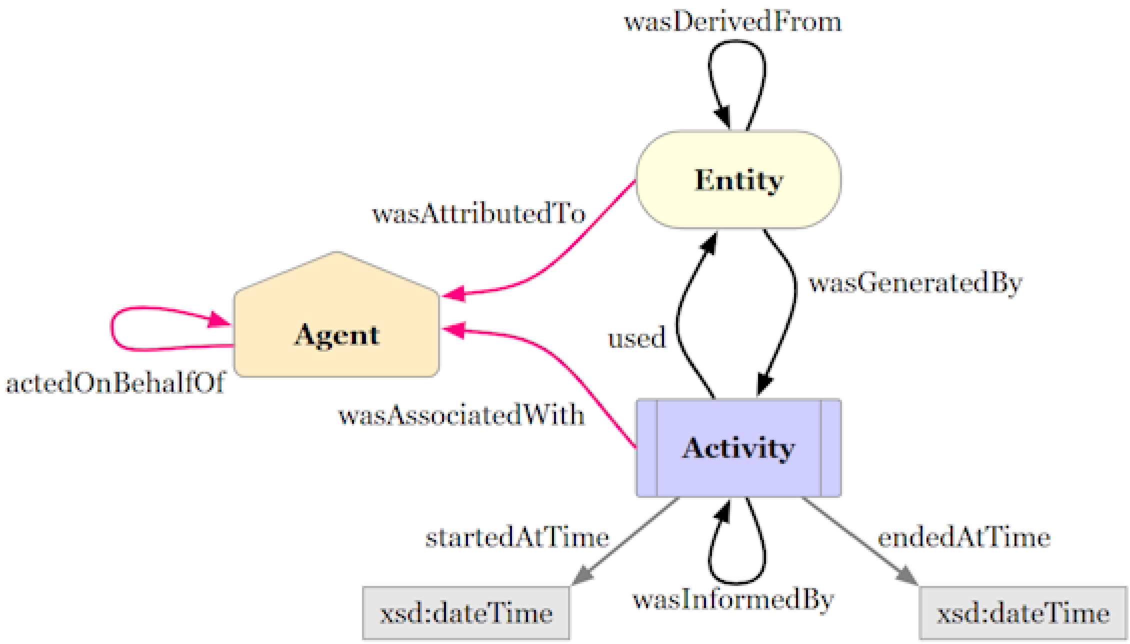

Figure 15 — Starting point classes from PROV-O Figure 1.

The PROV-O classes **Activity**, and **Entity** correspond almost exactly with:

1. BFO **process** (or its CCO subclass **Planned Act**), and **independent continuant**[18]; and

2. SSN **Execution**, and **Feature of Interest** or **Result** (W3C REC-vocab-ssn-20171019, W3C WD-vocab-ssn-2023-20250916)[19].

[18] PROV-BFO

[19] https://w3c.github.io/sdw-sosa-ssn/ssn#PROV-alignment

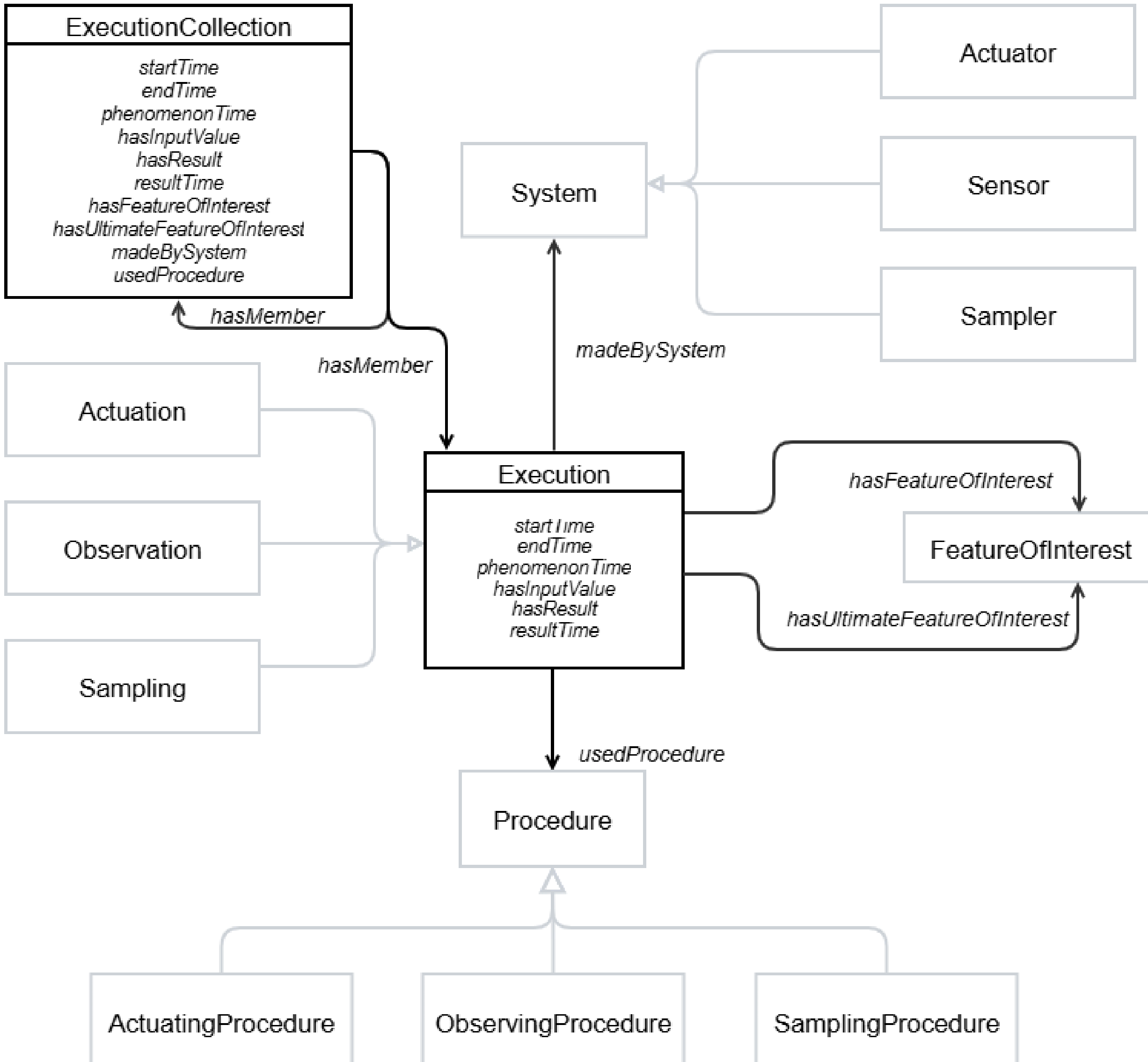

Figure 16 — SSN Execution overview

A summary of the SSN Ontology is provided in Annex D.

The base classes in the SSN Ontology (FeatureOfInterest, System, Procedure vs. Execution, Deployment) may be neatly fitted into BFO. Within CCO, the Event Ontology[20] defines several *kinds of* Planned Acts that may relate to imagery capture and processing, along with classes for Sensors, Actuators and Transducers that are used to accomplish this. Being both based on the fundamental disjuncture between continuants and occurrents, SSN and CCO are highly compatible.

[20] https://github.com/CommonCoreOntology/CommonCoreOntologies/blob/develop/src/cco-modules/EventOntology.ttl

A provisional alignment has been added to the new edition of the SSN Ontology[21].

A formal alignment, perhaps adding greater axiomatization to SSN will allow CCO-based data to link more easily with W3C/OGC-based data.

[21] https://w3c.github.io/sdw-sosa-ssn/ssn#BFO-alignment

5

RECOMMENDATIONS

5 RECOMMENDATIONS

Testbed-21 therefore makes the following recommendations for a coverage metadata ontology and thus for consideration in future releases of the IDO:

1. the ontology shall be based on the Common Core Ontology (CCO);
2. the ontology shall use GeoSPARQL for encoding spatial information;
3. the ontology shall have an explicit real-world referent for imagery; and
4. cell-properties shall be described using SWE Common JSON encoding.

In future

5. the sensor model and sensing chain should be modelled using a process-flow approach, consistent with the SSN Ontology and CCO Planned Act patterns
6. mappings to OGC API-Records, STAC and CIS must be developed, and pipelines deployed to allow IDO metadata to be catalogued in standard platforms.

Sections 3.1-3.3 and 4.2 elaborate on recommendations 1-4.

6

FUTURE WORK

6 FUTURE WORK

The ontology for coverage description proposed here covers part of the overall requirements for coverage metadata: specifically the elements required for geospatial registration, and the for the definition of the variable in the coverage range (e.g. pixel values).

In future work the following requirements should be addressed:

1. the sensor model and processing chain, including a formal mapping to SSN Ontology (W3C REC-vocab-ssn-20171019, W3C WD-vocab-ssn-2023-20250916);
2. a formal mapping to STAC (SpatioTemporal Asset Catalog, OGC 25-004);
3. a formal mapping to OGC API — Records (OGC 20-004r1);
4. a formal mapping to GeoDCAT (GeoDCAT-AP);
5. a formal mapping of the topology predicates in the CCO GeospatialOntology[5] to GeoSPARQL (OGC 22-047r1); and
6. profiles of the TB21 Ontology implemented as SHACL shapes (W3C REC-shacl-20170720).

7

SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

7 SECURITY, PRIVACY AND ETHICAL CONSIDERATIONS

The prototype TB21 ontology for coverage description is a OWL ontology. On top of the W3C OWL infrastructure, it is based on the following components that are already publicly available:

1. the Basic Formal Ontology (BFO, BFO-2020);
2. the GeospatialOntology and InformationEntityOntology modules from the Common Core Ontologies (CCO);
3. GeoSPARQL (OGC 22-047r1); and
4. SWE Common (OGC 24-014).

The TB21 ontology will be publicly available at https://github.com/opengeospatial/coverage-description-ontology .

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] STAC Community, Open Geospatial Consortium: OGC 25-004, *SpatioTemporal Asset Catalog (STAC) Community Standard*. Open Geospatial Consortium (2025). http://www.opengis.net/doc/CS/stac/1.1.

[2] Roger Lott, Open Geospatial Consortium: OGC 18-010r11, *Geographic information — Well-known text representation of coordinate reference systems*. Open Geospatial Consortium (2023). http://www.opengis.net/doc/is/crs-wkt/2.1.11.

[3] Katharina Schleidt, Ilkka Rinne, Open Geospatial Consortium: OGC 20-082r4, *Topic 20 — Observations, measurements and samples*. Open Geospatial Consortium (2023). http://www.opengis.net/doc/as/om/3.0.

[4] John Herring, Open Geospatial Consortium: OGC 06-103r4, *OpenGIS Implementation Specification for Geographic information — Simple feature access — Part 1: Common architecture*. Open Geospatial Consortium (2011). http://www.opengis.net/doc/is/sfa/1.2.1.

[5] Panagiotis (Peter) A. Vretanos, Tom Kralidis, Angelos Tzotsos, Charles Heazel, Open Geospatial Consortium: OGC 20-004r1, *OGC API — Records — Part 1: Core*. Open Geospatial Consortium (2025). http://www.opengis.net/doc/IS/ogcapi-records-1/1.0.

[6] Peter Baumann, Eric Hirschorn, Joan Masó, Open Geospatial Consortium: OGC 09-146r8, *OGC Coverage Implementation Schema with Corrigendum*. Open Geospatial Consortium (2019). http://www.opengis.net/doc/IS/cis/1.1.1.

[7] Nicholas J. Car, Open Geospatial Consortium: OGC 22-047r1, *OGC GeoSPARQL — A Geographic Query Language for RDF Data*. Open Geospatial Consortium (2024). http://www.opengis.net/doc/IS/geosparql/1.1.0.

[8] Alexandre Robin, Open Geospatial Consortium: OGC 24-014, *OGC SWE Common Data Model Encoding Standard*. Open Geospatial Consortium (2025). http://www.opengis.net/doc/IS/SWE/3.0.

[9] Butler H, Daly M, Doyle A, Gillies S, Hagen S, Schaub T, Internet Engineering Task Force: IETF RFC 7946, *The GeoJSON Format*. RFC Publisher (2016). https://www.rfc-editor.org/info/rfc7946.

[10] International Organization for Standardization (committee): ISO 19111:2019, *Geographic information — Referencing by coordinates*. International Organization for Standardization, Geneva (2019). https://www.iso.org/standard/74039.html.

[11] International Organization for Standardization (committee): ISO 19115-1:2014, *Geographic information — Metadata — Part 1: Fundamentals*. International Organization for Standardization, Geneva (2014). https://www.iso.org/standard/53798.html.

[12] International Organization for Standardization (committee): ISO 19125-1:2004, *Geographic information — Simple feature access — Part 1: Common architecture*. International Organization for Standardization, Geneva (2004). https://www.iso.org/standard/40114.html.

[13] International Organization for Standardization (committee): ISO/IEC 21838-2:2021, *Information technology — Top-level ontologies (TLO) — Part 2: Basic Formal Ontology (BFO)*. International Organization for Standardization, International Electrotechnical Commission, Geneva (2021). https://www.iso.org/standard/74572.html.

[14] World Wide Web Consortium (committee): W3C CR-rdf12-concepts-20260407, *RDF 1.2 Concepts and Abstract Data Model*. World Wide Web Consortium (2026). https://www.w3.org/TR/2026/CR-rdf12-concepts-20260407/.

[15] World Wide Web Consortium (committee): W3C NOTE-prov-overview-20130430, *PROV-Overview*. World Wide Web Consortium (2013). https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/.

[16] World Wide Web Consortium (committee): W3C REC-owl2-overview-20121211, *OWL 2 Web Ontology Language Document Overview (Second Edition)*. World Wide Web Consortium (2012). https://www.w3.org/TR/2012/REC-owl2-overview-20121211/.

[17] World Wide Web Consortium (committee): W3C REC-prov-o-20130430, *PROV-O: The PROV Ontology*. World Wide Web Consortium (2013). https://www.w3.org/TR/2013/REC-prov-o-20130430/.

[18] World Wide Web Consortium (committee): W3C REC-rdf-schema-20140225, *RDF Schema 1.1*. World Wide Web Consortium (2014). https://www.w3.org/TR/2014/REC-rdf-schema-20140225/.

[19] World Wide Web Consortium (committee): W3C REC-rdf11-concepts-20140225, *RDF 1.1 Concepts and Abstract Syntax*. World Wide Web Consortium (2014). https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/.

[20] World Wide Web Consortium (committee): W3C REC-shacl-20170720, *Shapes Constraint Language (SHACL)*. World Wide Web Consortium (2017). https://www.w3.org/TR/2017/REC-shacl-20170720/.

[21] World Wide Web Consortium (committee): W3C REC-skos-reference-20090818, *SKOS Simple Knowledge Organization System Reference*. World Wide Web Consortium (2009). https://www.w3.org/TR/2009/REC-skos-reference-20090818/.

[22] World Wide Web Consortium (committee): W3C REC-vocab-ssn-20171019, *Semantic Sensor Network Ontology*. World Wide Web Consortium (2017). https://www.w3.org/TR/2017/REC-vocab-ssn-20171019/.

[23] World Wide Web Consortium (committee): W3C REC-json-ld11-20200716, *JSON-LD 1.1*. World Wide Web Consortium (2020). https://www.w3.org/TR/2020/REC-json-ld11-20200716/.

[24] World Wide Web Consortium (committee): W3C REC-vocab-dcat-3-20240822, *Data Catalog Vocabulary (DCAT) — Version 3*. World Wide Web Consortium (2024). https://www.w3.org/TR/2024/REC-vocab-dcat-3-20240822/.

[25] World Wide Web Consortium (committee): W3C WD-vocab-ssn-2023-20250916, *Semantic Sensor Network Ontology — 2023 Edition*. World Wide Web Consortium (2025). https://www.w3.org/TR/2025/WD-vocab-ssn-2023-20250916/.

[26] Basic Formal Ontology (BFO) – https://basic-formal-ontology.org/bfo-2020.html

[27] Common Core Ontologies (CCO) – https://commoncoreontology.github.io/cco-webpage/

[28] DCAT-AP https://interoperable-europe.ec.europa.eu/collection/semic-support-centre/solution/dcat-application-profile-data-portals-europe

[29] Dublin Core Metadata Initiative (DCMI) Terms – https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

[30] The EPSG Geodetic Parameter Dataset http://www.epsg.org/

[31] The EPSG REST API https://apps.epsg.org/api/swagger/index.html

[32] GeoDCAT-AP https://semiceu.github.io/GeoDCAT-AP/

[33] GeoJSON-LD https://geojson.org/geojson-ld/

[34] JSON-LD context for CIS https://schemas.opengis.net/cis/1.1/rdf/

[35] JSON-Schema for CIS https://schemas.opengis.net/cis/1.1/json/

[36] JSON-Schema for SWE Common — https://schemas.opengis.net/sweCommon/3.0/json/

[37] Tim Prudhomme, Giacomo De Colle, Austin Liebers, Alec Sculley, Peihong "Karl" Xie, Sydney Cohen, and John Beverley, A semantic approach to mapping the Provenance Ontology to Basic Formal Ontology — https://arxiv.org/pdf/2408.03866

[38] QUDT — Quantities, Units, Dimensions and Data Types Ontologies — https://www.qudt.org/

[39] Randell, D. A., Cui, Z. and Cohn, A. G.: 1992, "A spatial logic based on regions and connection," Proc. 3rd Int. Conf. on Knowledge Representation and Reasoning, Morgan Kaufmann, San Mateo, pp. 165-176 — https://www.dpi.inpe.br/gilberto/references/cohn_rcc.pdf

[40] SpatioTemporal Asset Catalog (STAC) specification — https://stacspec.org/en

[41] The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. Krzysztof Janowicz; Michael Compton. CEUR: Proceedings of the 3rd International Workshop on Semantic Sensor Networks (SSN10). 2010 — http://ceur-ws.org/Vol-668/paper12.pdf

[42] Clementini, Eliseo; Di Felice, Paolino and van Oosterom, Peter: A small set of formal topological relationships suitable for end-user interaction (1993). In Abel, David; Ooi, Beng Chin (eds.). Advances in Spatial Databases: Third International Symposium, SSD ’93 Singapore, June 23–25, 1993 Proceedings. Lecture Notes in Computer Science. 692/1993. Springer. pp. 277–295. https://doi.org/10.1007/3-540-56869-7_16

A

ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

A

ANNEX A (NORMATIVE) ABBREVIATIONS/ACRONYMS

BFO	Basic Formal Ontology (ISO)
CCO	Common Core Ontologies
CIS	Coverage Implementation Schema (OGC)
CRS	Coordinate Reference System
DCAT	Data Catalog Vocabulary (W3C)
DCAT-AP	Data Catalogue Application Profile (SEMIC.EU)
DCMI	Dublin Core Metadata Initiative
DCterms	Dublin Core Metadata Terms (DCMI)
ECMA	European Computer Manufacturers Association
GeoJSON	Geographic JSON (IETF)
GeoSPARQL	Geographic SPARQL — A Geographic Query Language for RDF Data (OGC)
IDO	Imagery Domain Ontology
IETF	Internet Engineering Task Force
ISO	International Organization for Standardization
JSON	Javascript Object Notation (IETF)
JSON-LD	JSON for Linked Data (W3C)
OGC	Open Geospatial Consortium
OWL	Web Ontology Language (W3C)
RDF	Resource Description Framework (W3C)
RDFS	RDF Schema (W3C)

SEMIC.EU	The Semantic Interoperability Centre Europe
SHACL	Shapes Constraint Language (W3C)
SKOS	Simple Knowledge Organization System (W3C)
STAC	SpatioTemporal Asset Catalog (OGC)
TTL	Turtle — Terse RDF Triple Language (W3C)
UML	Unified Modeling Language
W3C	World Wide Web Consortium
WGS84	World Geodetic System 1984
WKT	Well-Known Text (ISO, OGC)

B

ANNEX B (INFORMATIVE) SUMMARY OF THE PROPOSED ONTOLOGY FOR COVERAGE DESCRIPTION

ANNEX B (INFORMATIVE) SUMMARY OF THE PROPOSED ONTOLOGY FOR COVERAGE DESCRIPTION

B.1. GeoSPARQL-BFO alignment

Table B.1 — GeoSPARQL-BFO alignment axioms

Classes			
:Feature[a]	≡	“independent continuant” **or** “process”[b]	*Anything that might be shown on a map*
:Feature	⊆	[:hasGeometry **some** sf:Geometry[c]][d]	*Use simple-features Geometry types*
:Feature	⊆	[:hasBoundingBox **some** sf:Envelope[e]]	
:Geometry	⊆	“information content entity”	
“spatial region”	⊆	:Feature	*BFO “immaterial entities” may carry :hasGeometry and :hasBoundingBox*
“site”	⊆	:Feature	
“continuant fiat boundary”	⊆	:Feature	
Object Properties			
:hasBoundingBox	⊆	“externally connects with”	
:hasGeometry	⊆	“described by”	

[a] An unqualified compact URI is a GeoSPARQL term

[b] A quoted label is a BFO, CCO, or TB21 term

[c] sf:Geometry is a formal subclass of :Geometry

[d] An [expression in square brackets] is an owl:Restriction

[e] sf:Envelope is a formal subclass of sf:Geometry

The axiomatization of the GeoSPARQL-BFO alignment is packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/blob/main/ontology/tb21-region.ttl

B.2. Region of interest definition

Table B.2 — Region of interest axioms

Classes			
“spatial region”	⊆	:Feature	
“region of interest”	⊆	“spatial region”	*Specialization of BFO class*
“region of interest”	⊆	[“has boundary” **only** “region boundary”]	
“region of interest”	⊆	[:hasBoundingBox **some** sf:Envelope]	*From :Feature*
“region of interest”	⊆	[:hasGeometry **some** sf:Geometry]	
“region boundary”	⊆	“continuant fiat boundary”	*Specialization of BFO class*
“region boundary”	⊆	[“is boundary of” only “region of interest”]	
“region boundary”	⊆	[“has vertex” **min** 2]	
“vertex”	⊆	“fiat point”	*Specialization of BFO class*
“vertex”	⊆	[“is vertex of” **only** “continuant fiat boundary”]	

“vertex”	⊆	[:hasGeometry **some** sf:Point]	
Object properties			
“has boundary”	⊆	“has tangential part”	*Specialization of CCO object property*
“has boundary”	**domain**	“spatial region”	
“has boundary”	**range**	“continuant fiat boundary”	
“has vertex”	⊆	“has tangential part”	*Specialization of CCO object property*
“has vertex”	**domain**	“continuant fiat boundary”	
“has vertex”	**range**	“vertex”	

The axiomatization of `region of interest` is packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/blob/main/ontology/tb21-region.ttl

B.3. Image description

Table B.3 — Image axioms

Classes			
“Image (of a region)”	⊆	“Representational Information Content Entity”	*Specialization of CCO class*
“Image (of a region)”	⊆	[“represents” only “region of interest”]	
“value specification”	⊆	“Information Content Entity”	*Specialization of CCO class*
“TAI timestamp in ns”	⊆	“value specification”	
“TAI timestamp in ns”	⊆	[“has unit” **value** “ns”]	
“TAI timestamp in ns”	⊆	[“uses reference system” **value** “International Atomic Time”]	

“coordinate”	⊆	“value specification”	
“raster coordinate”	⊆	“coordinate”	
“CCS coordinate”	⊆	“raster coordinate”	
“CCS coordinate”	⊆	[“has column number” **some** xsd:integer]	
“CCS coordinate”	⊆	[“has row number” **some** xsd:integer]	
“raster element correspondence”	⊆	“Information Content Entity”	*Specialization of CCO class*
“raster element correspondence”	⊆	[“has correspondence source” **min** 1]	
“raster element correspondence”	⊆	[“has correspondence target” **min** 1]	
“raster element ground correspondence”	⊆	“raster element correspondence”	
“raster element ground correspondence”	⊆	[“has correspondence source” **exactly** 1 “CCS coordinate”]	
“raster element ground correspondence”	⊆	[“has correspondence target” **exactly** 1 sf:Point]	
“corner correspondence”	⊆	“raster element ground correspondence”	
“RangeType”	⊆	“Prescriptive Information Content Entity”	*Specialization of CCO class*
“RangeType”	⊆	[“has DataRecord” **max** 1]	
Object Properties			
“has timestamp”	**range**	“value specification”	
“has information content entity part”	**domain**	“Information Content Entity”	
“has information content entity part”	**range**	“Information Content Entity”	
“has corner”	⊆	“has information content entity part”	
“has corner”	**domain**	“Image (of a region)”	
“has corner”	**range**	“CCS coordinate”	
“has correspondence source”	⊆	“has information content entity part”	

“has correspondence source”	**domain**	“raster element correspondence”
“has correspondence source”	**range**	[“coordinate” **and** (“uses reference system” **some** “raster image coordinate reference system”)]
“has correspondence target”	⊆	“has information content entity part”
“has correspondence target”	**domain**	“raster element correspondence”
“has correspondence target”	**range**	“fiat point”
“has cell property type”	⊆	“prescribed by”
“has cell property type”	**domain**	“Image (of a region)”
“has cell property type”	**range**	“RangeType”
Data Properties		
“has column number”	**range**	xsd:integer
“has row number”	**range**	xsd:integer
“has DataRecord”	**domain**	“RangeType”
“has DataRecord”	**range**	rdf:JSON

The axiomatization of `Image` is packaged in this ontology file

- https://github.com/opengeospatial/coverage-description-ontology/blob/main/ontology/tb21-image.ttl

C

ANNEX C (INFORMATIVE) SOME VOCABULARY AND ONTOLOGY CONCEPTS

ANNEX C (INFORMATIVE) SOME VOCABULARY AND ONTOLOGY CONCEPTS

C.1. RDF vocabularies

RDF (W3C CR-rdf12-concepts-20260407) is a *graph data model*. Every node and arc in an RDF graph or dataset is denoted by a IRI, except for the literals (strings, numbers, etc) on terminal nodes.

Vocabularies of terms for RDF applications may be defined using RDFS (W3C REC-rdf-schema-20140225), OWL (W3C REC-owl2-overview-20121211), SHACL (W3C REC-shacl-20170720), or a combination of these.

Each term in an RDF vocabulary is denoted by a IRI. The IRI provides a link to the original definition. Groups of terms from the same source typically have the same IRI stem, known as a *namespace*.

It is thus possible to combine terms and vocabularies from different sources. Applications are often built from terms in multiple namespaces.

C.2. Ontologies

An Ontology is a set of terms, axioms, and definitions, and the relationships within it, defined with formal logic, using OWL (W3C REC-owl2-overview-20121211). The primary purpose of ontologies is to support reasoning and inference.

Ontologies, are usually implemented in RDF. However, use of RDF does not require an ontology.

C.3. Ontology vs Metadata Vocabulary

Is it necessary to use an Ontology to define a metadata vocabulary? It depends.

For applications that depend on the use of reasoning and inference, adoption of a consistent, formal ontology is essential.

However, use of OWL ontologies in other contexts can impose significant burdens.

1. The design process can be onerous.
2. Alignment with standards already in use in the community can be difficult.

If compatibility with existing software is desirable, and the application can be serviced with descriptive metadata generated in a controlled environment, then an RDF vocabulary utilising existing standards may serve the purpose.

A formal ontology should only be considered if the application involves reasoning.

C.4. RDF Validation

SHACL (W3C REC-shacl-20170720) is a W3C standard for describing constraints on RDF graphs. SHACL can specify the selection of terms from RDF vocabularies (namespaces) and their arrangement and obligations within a dataset, such as a metadata record.

SHACL can be used to define the structure of an RDF document, either in addition to or even *in place of* OWL. In particular, unlike OWL and RDFS, which operate under the Open World Assumption and deal with entailments and support analysis of *consistency*, SHACL can be used to define *content validity rules* for an application.

C.5. RDF serialization

An RDF instance ("graph") can be serialized as XML, Turtle (TTL), JSON, N3, TriG, and other formats.

JSON-LD (W3C REC-json-ld11-20200716) is the JSON-based format for RDF. The JSON-LD "Context" is a header which converts the JSON keys to IRIs that denote RDF predicates (arc names), and thus allows (some) JSON documents to be interpreted as valid RDF instances.

D

ANNEX D (INFORMATIVE) EXISTING STANDARDS RELATED TO GEOSPATIAL METADATA AND DATA

D

ANNEX D (INFORMATIVE) EXISTING STANDARDS RELATED TO GEOSPATIAL METADATA AND DATA

This section briefly describes some existing standards that may be relevant to the scope of the project.

D.1. OGC API — Records

OGC-API-Records (OGC 20-004r1) is the standard OGC REST API for geospatial metadata records. Records are typically serialized in GeoJSON: see IETF RFC 7946. The 'core properties' functionally provides a metadata model (#core-properties).

Table D.1 — Core properties related to the resource

PROPERTY	REQUIREM	DESCRIPTION	GEOJSON KEY
`type`	optional	The nature or genre of the resource described by this record.	`properties.type`
`title`	optional	A human-readable name given to the resource described by this record.	`properties.title`
`description`	optional	A free-text description of the resource described by this record.	`properties.description`
`geometry`	optional	A spatial extent associated with the resource described by this record. Can be null if there is no associated spatial extent.	`geometry`
`time`	optional	A temporal extent associated with the resource described by this record. Can be null if there is no associated temporal extent.	`time`
`keywords`	optional	A list of free-form keywords or tags associated with the resource described by this record.	`properties.keywords`
`themes`	optional	A knowledge organization system used to classify the resource described by this resource.	`properties.themes`

PROPERTY	REQUIREN	DESCRIPTION	GEOJSON KEY
`resourceLanguages`	optional	The list of languages in which the resource described by this record can be retrieved.	`properties.resourceLanguages`
`externalIds`	optional	One or more identifiers, assigned by an external entity, for the resource described by this record.	`properties.externalIds`
`formats`	optional	A list of available distributions for the resource described by this record.	`properties.formats`
`contacts`	optional	A list of contacts qualified by their role(s) in association to the record or the resource described by this record.	`properties.contacts`
`license`	optional	The legal provisions under which the resource described by this record is made available.	`properties.license`
`rights`	optional	A statement that concerns all rights not addressed by the license such as a copyright statement.	`properties.rights`

The GeoJSON-LD context lifts GeoJSON to RDF. *In principle* this may allow the core properties of the OGC API — Records standard to provides a geospatial metadata model implemented in RDF.

D.2. SpatioTemporal Asset Catalog (STAC)

D.2.1. STAC

SpatioTemporal Asset Catalog (SpatioTemporal Asset Catalog, [STAC_OGC]) provides a common structure for describing and cataloging spatiotemporal assets. A spatiotemporal asset is any file that represents information about the earth captured in a certain space and time particularly satellite imagery, SAR, lidar.

STAC has been deployed globally by many significant earth observation imagery providers:

- NASA Earthdata
- Copernicus data space ecosystem
- Digital Earth Africa

STAC is recognised as a Community Standard by OGC (OGC 25-004).

STAC Item is the core atomic unit, which describes a single spatiotemporal asset, such as an image, model, or video. A STAC Item is implemented as a GeoJSON feature plus datetime and links.

This object describes a STAC Item. The fields `id`, `type`, `bbox`, `geometry` and `properties` are inherited from GeoJSON.

Table D.2 — STAC Item

FIELD NAME	TYPE	DESCRIPTION
`type`	string	REQUIRED. Type of the GeoJSON Object. MUST be set to Feature.
`stac_version`	string	REQUIRED. The STAC version the Item implements.
`stac_extensions`	[string]	A list of extensions the Item implements.
`id`	string	REQUIRED. Provider identifier. The ID should be unique within the Collection that contains the Item.
`geometry`	GeoJSON Geometry Object | null	REQUIRED. Defines the full footprint of the asset represented by this item, formatted according to RFC 7946, section 3.1 if a geometry is provided or section 3.2 if no geometry is provided.
`bbox`	[number]	REQUIRED if geometry is not null, prohibited if geometry is null. Bounding Box of the asset represented by this Item, formatted according to RFC 7946, section 5.
`properties`	Properties Object	REQUIRED. A dictionary of additional metadata for the Item.
`links`	[Link Object]	REQUIRED. List of link objects to resources and related URLs. See the best practices for details on when the use self links is strongly recommended.
`assets`	Map<string, Asset Object>	REQUIRED. Dictionary of asset objects that can be downloaded, each with a unique key.
`collection`	string	The id of the STAC Collection this Item references to. This field is required if a link with a collection relation type is present and is not allowed otherwise.

The GeoJSON-LD context lifts GeoJSON to RDF. *In principle* this may allow a STAC Item to be interpreted as RDF.

D.2.1.1. Instrument

https://docs.ogc.org/cs/25-004/25-004.html#instrument

D.2.1.2. Bands

https://docs.ogc.org/cs/25-004/25-004.html#bands

D.2.1.3. Data

null-values, units etc

https://docs.ogc.org/cs/25-004/25-004.html#data-values

D.3. DCAT, GeoDCAT

DCAT (W3C REC-vocab-dcat-3-20240822) is a W3C standard for data catalogs. It is designed using RDFS and OWL for implementation in RDF.

The core class `dcat:Dataset` describes a dataset, which is "a collection of data, published or curated by a single agent, and available for access or download in one or more formats" (#Class:Dataset). An image or video can be considered a dataset.

DCAT uses a large set of properties to describe the dataset, such as title, description, keyword, publisher, etc. Many of these are from DCterms (Dublin Core terms) in the `dcterms:` namespace. Additional properties are defined in the `dcat:` namespace such as `dcat:distribution`, `dcat:inSeries`, `dcat:qualifiedRelation`, or taken from other standard namespaces such as `prov:wasGeneratedBy`. Basic support for time and space uses `dcterms:spatial`, `dcat:spatialResolutionInMeters`, `dcterms:temporal`, `dcat:temporalResolution` (#time-and-space).

GeoDCAT-AP ([GeoDCAT-AP]) is a profile of DCAT that adds more elements for geographic information, and defines specific patterns compatible with the expectations of the geospatial community, particularly as embodied in ISO 19115 (ISO 19115-1:2014). GeoDCAT-AP is based on the DCAT_AP ([DCAT_AP]) application profile for data portals in Europe.

A generic (non-European) version of GeoDCAT is under development by OGC.

D.4. GeoSPARQL

GeoSPARQL (OGC 22-047r1) is the OGC standard for representing geospatial data in RDF.

- geo:Feature
- geo:hasBoundingBox
- geo:hasGeometry
- geo:Geometry

- geo:hasSerialization
- geo:hasWKT
- geo:wktLiteral

D.4.1. GeoSPARQL classes

GeoSPARQL provides an RDF implementation of the basic concepts *Feature* and *Geometry* from the GIS community.

The GeoSPARQL property `hasGeometry` links a `Feature` to its `Geometry`.

The GeoSPARQL property `hasSerialization` links a `Geometry` to a representation as a set of coordinates or some other notation.

D.4.2. GeoSPARQL geometry literals

Serialization of geometry in GeoSPARQL is in formatted literals. GeoSPARQL supports the following formats: GML, WKT, GeoJSON, KML, and DGGS (OGC 22-047r1, Section 10.8).

D.4.3. GeoSPARQL spatial relationships

GeoSPARQL includes terms for spatial or topological relationships. Three sets are provided, corresponding to usage in the geospatial community:

- Egenhofer-Herring spatial relations
- Simple Features spatial relations
- RCC8 spatial relations

Rigorous definitions of each relation in terms of its DE-9IM pattern is provided.

D.5. GeoJSON

GeoJSON (IETF RFC 7946) is a lightweight standard for representing geospatial data in JSON format. GeoJSON was designed *pragmatically* to support transfer of spatial data for web applications. It is not intended to be a complete, formal standard for geospatial data.

However, the compact GeoJSON notation for 2-D geometries is widely supported in web software, and might be used in other contexts. GeoJSON encodes geometry in arrays of numbers. For example:

```
{
    "type": "Point",
    "coordinates": [145.112566, -37.848910]
}
```

Listing D.1

GeoJSON geometries are restricted to *decimal degrees* in *WGS84*.

GeoJSON-LD (GeoJSON-LD) provides a JSON-LD Context for GeoJSON, which allows GeoJSON documents to be interpreted as valid RDF instances.

D.6. OGC CIS

OGC Coverage Implementation Schema CIS (OGC 09-146r8) defines how to model coverages — data that vary over space and/or time, such as satellite imagery, climate data, or elevation grids. CIS is specified in UML, with various implementations. The representation of gridded data is summarized in Section 25 and Figure 5.

The description of the *variables* in an image or other coverage is the CIS *range type*. The rangetype-context provides a convenient overview of its scope and structure. The details of the record type and structure are delegated in turn to [SWE_Common].

CIS v1.1 provides a JSON-LD encoding. Example for grids are available at https://github.com/opengeospatial/coverage-implementation-schema/tree/main/standard/schemas/1.1/json/examples/generalGrid. In particular 2D_regular.json shows how image indexes (“generalGrid/gridLimits/axis type=IndexAxis”) are mapped to ground coordinates (“generalGrid/axis type=RegularAxis”).

D.7. Semantic Sensor Network Ontology

The SSN Ontology Sensor (W3C REC-vocab-ssn-20171019, W3C WD-vocab-ssn-2023-20250916) provides the canonical RDF implementation of OGC’s Observations, Measurements and Samples (OGC 20-082r4) standard (also known as ISO 19156:2023).

In SSN a `Sensor` is a subClass of `System`. A System implements a `Procedure` in order to make an `Execution`.

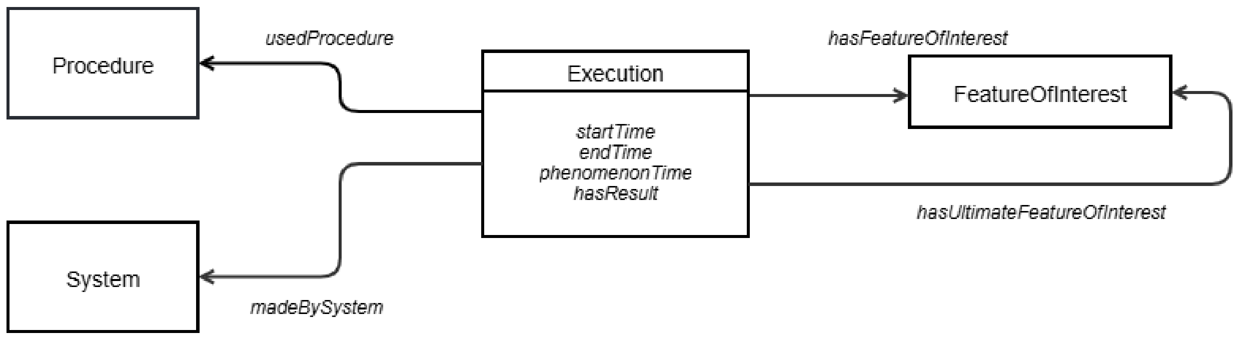

Figure D.1 — Overview of Execution pattern

A Sensor implements an `ObservingProcedure` in order to make an `Observation`. The result of an Observation is information — specifically, an estimate of the value of a `Property` of a `Feature of Interest`. Individual Sensor observations are triggered by a `Stimulus` (Stimulus-Sensor-Observation Ontology).

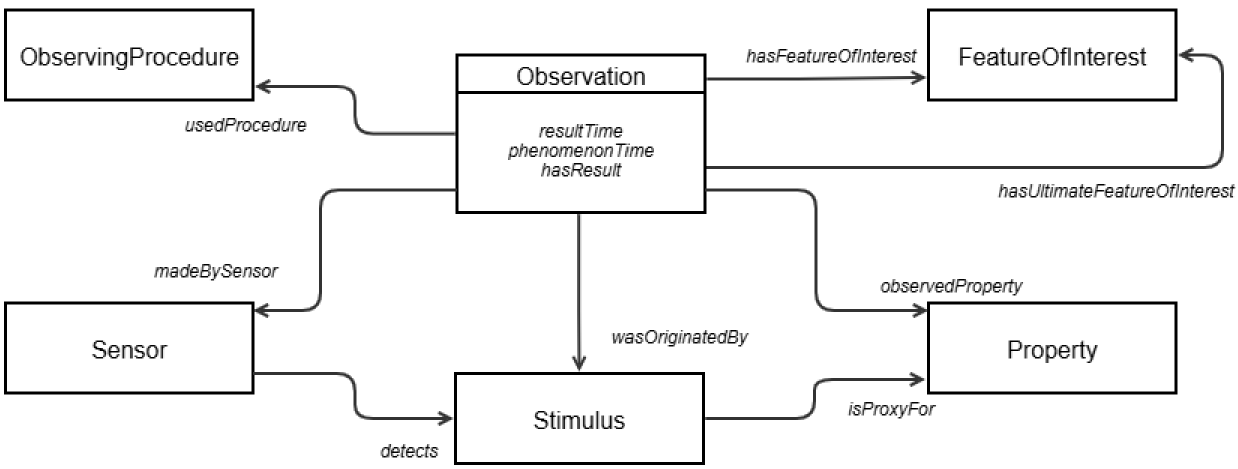

Figure D.2 — Overview of Observation pattern

The SSN Ontology has a single Sensor class. A System (Sensor) may be mounted on a `Platform` in the context of a `Deployment`. A System may be composed of sub-systems. Otherwise, SSN does not prescribe finer details of Sensor description or specification.

Non-normative examples and patterns for describing sensors composed of subsystems are documented in W3C REC-vocab-ssn-20171019, W3C WD-vocab-ssn-2023-20250916, but these are limited in scope.

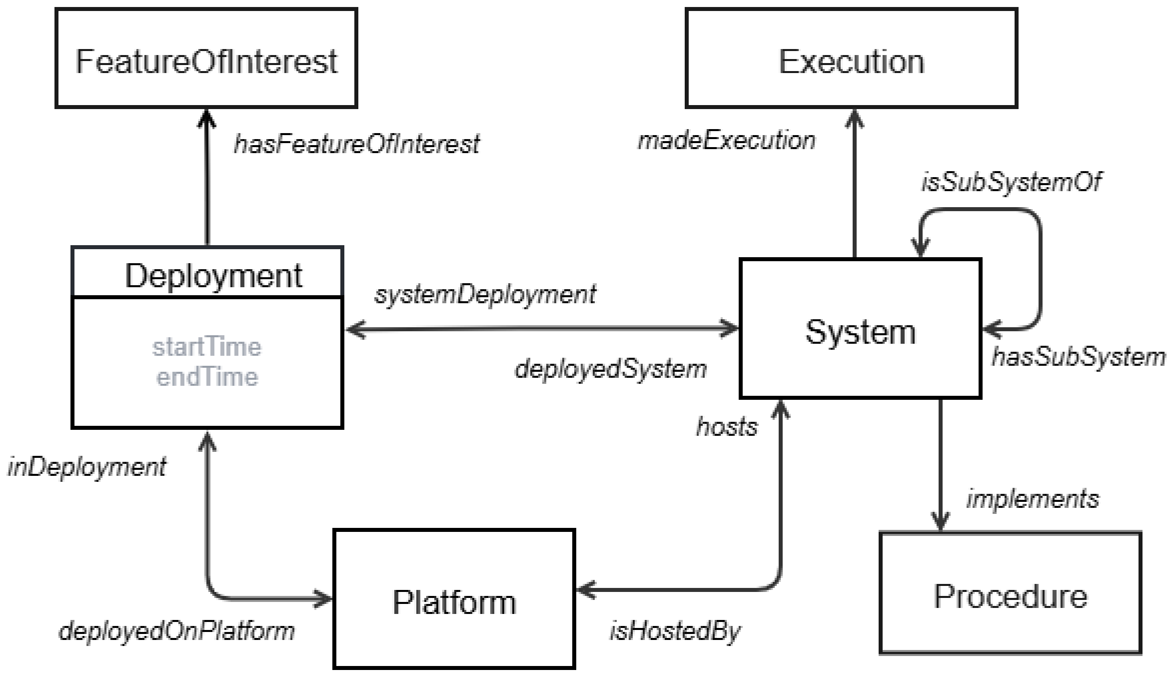

Figure D.3 — Overview of Deployment pattern

SSN was not designed in the BFO framework — it is a standalone ontology, with dependencies on W3C REC-owl2-overview-20121211/[RDF]/W3C REC-skos-reference-20090818 from the W3C framework, and on QUDT.

The fundamental design is highly compatible with BFO: `Observation` is an `occurrent` (`process`), which generates an `information object` that describes some `quality` of a (usually) `independent continuant`. ("Usually" is because while the feature-of-interest is most often a static entity in the world, it could also be a `process` or maybe some other kind of `occurrent`.)