Coverages, JSON-LD and RDF data cubes

Conference or Workshop Item

Published Version

Blower, J. and Riechert, M. (2016) Coverages, JSON-LD and RDF data cubes. In: Workshop on Spatial Data on the Web (SDW 2016), 27-30 Sep 2016, Montreal, Canada. Available at http://centaur.reading.ac.uk/74395/

Published version at: http://ceur-ws.org/Vol-1777/

Coverages, JSON-LD and RDF Data Cubes

J.D. Blower* and M. Riechert

Department of Meteorology, University of Reading, United Kingdom

Abstract. Many kinds of scientific data, including satellite imagery, climate simulations and sensor data, can be represented as *coverages*, which are essentially mappings from points in space and time to data values. Coverage data are typically encoded as multidimensional arrays in compact binary forms such as NetCDF, HDF and GeoTIFF, most of which require specialist knowledge and tools to process and manipulate. There is considerable current interest in helping the wider Web community use coverage data, by providing data in more general formats such as JSON and RDF, and by using commonly-accepted vocabularies. This short discussion paper outlines some current work in the area and highlights some of the main inherent issues.

Keywords: Coverages, RDF, JSON-LD, Semantic Web, efficiency

1 Introduction and Motivation

A *coverage* is a general data structure that can hold many different kinds of geospatial and temporal data. It is defined in the ISO19123 specification [7] as a "feature that acts as a function to return values from its range for any direct position within its spatial, temporal or spatiotemporal domain". Essentially, a coverage maps a set of positions in space and time to data values. Examples of coverage data include satellite imagery, results from simulations of weather and climate, flow data from river gauges, wind observations from radiosondes and much more. It is important to note that coverages can be gridded or non-gridded in nature (the term "coverage" is sometimes used as a synonym for "gridded" or "raster" data but this is not the case).

Coverages are typically published on the Web in one of two ways: (i) as static files, in formats such as GeoTIFF, NetCDF [13] and HDF [6]; or (ii) through web APIs such as the Web Coverage Service (WCS) [15] or OPeNDAP [5]. Users from outside a selection of specialist communities often find it hard to access and understand coverage data, for reasons including complexity, unfamiliarity and large data size.

The user base of coverage data could potentially be widened if it were possible to make data available in more general formats that are familiar to the Web and Semantic Web communities. In addition, the use of well-defined vocabularies would help to describe coverage data better, and help it to be related more

---

* j.d.blower@reading.ac.uk

accurately to other related data sources. Therefore, it is interesting to consider how coverage data could be encoded in RDF [2], and to what extent this is practical and usable.

The joint OGC/W3C "Spatial Data on the Web" working group[1] is developing several strands of work in this area, and this work has been partly motivated by the activities of this group. This short paper highlights some of the issues revealed, with the intention of stimulating discussion. It is not intended to be a complete treatment of this complex area, nor does it represent the totality of the work of the group.

2 The main elements of a Coverage

A useful general structure for a Coverage can be found in the Open Geospatial Consortium Coverage Implementation Schema (CIS) [3], which specialises ISO19123. In summary, a coverage consists of three main pieces of information:

- The *domain*, which encodes the set of points in space and time for which data values exist.
- The *range*, which is the set of data values themselves.
- The *range metadata*, which describes the range values (e.g. units of measure).

The only structural difference between different types of coverage (images, timeseries, vertical profiles etc.) is the spatiotemporal geometry and topology of the domain. The range and range metadata can be encoded consistently, irrespective of the domain type.

(We note that an alternative model is possible, in which the domain and range are interleaved. This model can be convenient in some cases – for example where streaming capabilities are required – but we do not discuss it further in this paper.)

3 CoverageJSON and JSON-LD

CoverageJSON (or "CovJSON" for short) is a JSON-based [1] format that we have developed for encoding coverage data. It is primarily intended to be used as a means to convey data from servers to web browsers, so that web developers can develop rich, highly interactive applications based on coverage data. The primary goals of CovJSON are usability and efficiency. The format is described more fully at https://covjson.org, but notable features include:

- The ability to encode a large range of domain types (grids, timeseries, trajectories, polygonal representations, and more).
- The ability to separate information between documents, which are linked together (e.g. the range can be held in a separate document), enabling more convenient treatment of large data sets.

---

[1] `https://www.w3.org/2015/spatial/`

– The use of URIs to encode semantic information.

A sample CovJSON document is given in Appendix A.

CovJSON makes use of JSON-LD [9], a means of representing RDF data in JSON. Through the use of a JSON-LD context file, the objects and properties in a CovJSON file can be converted to URIs and triples. An important “tension” in our development of CovJSON has been balancing the desire to use JSON-LD (to enable automated conversion to RDF) with the desire to maintain the readability and simplicity of the format. We believe that it is of great importance that users should be able to use and understand CovJSON without the need to understand JSON-LD or RDF – hence we have used simple and idiomatic JSON as our starting point, using JSON-LD to enable some degree of interoperability with RDF, accepting the limitations of doing so for now.

The result of converting the CovJSON document in Appendix A to RDF, using the information in the currently-supplied JSON-LD context document, is shown in Appendix B. It can be seen that the range metadata (coming from the CovJSON Parameter objects) is well represented in the resulting RDF, as well as the domain type (a two-dimensional grid in this case) and certain domain metadata such as the coordinate reference systems (CRSs). However, the actual data values that make up the domain and range do not survive the conversion. Other information is also missing, such as the mapping between the axes of the domain and the coordinate reference systems.

The two main reasons for this are:

1. Much of a coverage’s data is array-based in CovJSON, including the range and key parts of the domain. JSON arrays can be converted to RDF lists through JSON-LD, but this is probably not a usable way to represent arrays that can potentially be very large.
2. CovJSON uses data structures that are equivalent to associative arrays. The keys of these arrays are important and are used to map between different pieces of information in the CovJSON document. For example, the arrays representing each variable in the coverage are keyed to short identifier strings, which are also used in Parameters – see the use of “`TEMP`” in Appendix A. These keys cannot be easily converted to an RDF representation with the current version of JSON-LD (a change to JSON-LD has been requested and may appear in future versions [2]). One potential solution is to craft a different JSON-LD context document for each CovJSON document, but we feel it would be much more desirable to use a single, standard context document that works across all CovJSON documents.

We therefore conclude that, for the foreseeable future, the automated “full” conversion of CovJSON to RDF will remain limited. However, some applications may find the resulting limited RDF useful, particularly for describing the range metadata and the types of the main objects in the coverage. Applications that do not require RDF will hopefully appreciate the simple nature of the format and the use of URIs to bring in some semantic content.

---

[2] https://github.com/json-ld/json-ld.org/issues/430

4 DataCube

The RDF Data Cube vocabulary [12] provides an general means for encoding multidimensional data. Although not specifically designed for representing coverage data, coverages are certainly (in general) multidimensional objects that share many of the characteristics of a Data Cube. We have not yet attempted to encode coverage data using the Data Cube vocabulary, but are aware of efforts ongoing elsewhere [11]. We would therefore like to stimulate discussion about whether the Data Cube is the most useful RDF representation of a coverage. If this is the case, then we may be able to treat the Data Cube representation as the "canonical" RDF coverage encoding and provide tools for converting CovJSON to Data Cubes. There should be enough information in the CovJSON document to be able to perform this conversion unambiguously.

In fact, the way in which CovJSON models coverages is conducive to such a conversion. In CovJSON, all types of domain are modelled using a set of orthogonal axes, even where the domain consists of polygons. Each of these axes could map neatly to a dimension in the data cube. (It is worth noting that CIS does not always model domains using orthogonal axes, and so conversion from CIS to Data Cube may be less obvious in some cases.)

The Data Cube vocabulary is designed to encode statistical data, in which the dimensions of the data cube are relatively small and unordered. The axes of a coverage can be large (satellite images can be tens of thousands of pixels across) and ordering of the axis values is important. Therefore the Data Cube vocabulary may require some specialisation to reflect this.

As with CovJSON and JSON-LD, the encoding of the range values (particularly for large coverages) will be an important issue to resolve. Possible solutions include:

1. encode the range values individually as RDF triples, which will be very inefficient for large coverages;
2. define a microformat (i.e. encode all range values in a single long string). This will be more efficient, but less flexible and still problematic for large coverages; or
3. encode the range values in a separate document in a more efficient format.

The last option is likely to be the most productive. Various formats for the range are possible, but one attractive possibility is CovJSON itself, which provides a JSON encoding for a standalone multidimensional array (this can be compressed during data transfer for much greater efficiency). It may also, of course, be possible to use binary formats like NetCDF for this purpose. But many of these formats encode the full coverage (not just the range), and care must be taken to ensure that the RDF representation of the domain is consistent with that in the linked file.

5 Summary

RDF can provide a highly expressive means to encode coverage information. Through the use of other vocabularies that provide contextual information (such as DCAT [4] for dataset-level descriptions and PROV [10] for provenance information) it may become possible to describe the whole “coverage ecosystem” consistently in RDF, which would be a great benefit to data managers and users. Much work needs to be done to work out how these vocabularies should work with each other, and how systems like Observations and Measurements [8] can best be used.

CovJSON provides a general data model for a wide range of coverages (based on CIS concepts), making significant use of semantic information. It defines a kind of ontology for coverages, although currently not a complete one. The incompleteness is partly due to the current limitations of the current version of JSON-LD. The RDF Data Cube may provide a more useful “canonical” RDF format, and we are hopeful that CovJSON and DataCube representations are sufficiently conceptually similar that they can be converted readily to each other. This would be an interesting area of research.

The Semantic Sensor Network (SSN) [14] provides another abstract model that could probably be specialised or extended to act as a representation of coverages. Like the RDF Data Cube, it would need to be endowed with concepts of things like dimensions, coordinate reference systems and so forth. This could be an interesting area of discussion and research.

It is likely that any attempt to encode coverages as RDF will meet practical limitations in encoding the range values (and, in some cases, the domain values). Given that range values will probably be given in separate documents outside the “RDF world” it is critical that the RDF and non-RDF information can be related unambiguously to each other.

References

1. Bray, T.: The JavaScript Object Notation (JSON) Data Interchange Format. RFC 7159, RFC Editor (March 2014), `http://www.rfc-editor.org/rfc/rfc7159.txt`
2. Candan, K.S., Liu, H., Suvarna, R.: Resource Description Framework: Metadata and Its Applications. SIGKDD Explor. Newsl. 3(1), 6–19 (Jul 2001)
3. Coverage Implementation Schema. Standard 09-146r2, Open Geospatial Consortium (May 2012), `http://www.opengis.net/doc/gml/gmlcov/1.0.1`
4. Data Catalog Vocabulary (DCAT). Standard, W3C (Jan 2014), `https://www.w3.org/TR/vocab-dcat/`
5. Gallagher, J., Potter, N., Sgouros, T., Hankin, S., Flierl, G.: The Data Access Protocol - DAP 2.0. Standard ESE-RFC-004.0.08, NASA (October 2005)
6. Hierarchical Data Format, version 5. Standard, The HDF Group (1997-2016), `https://support.hdfgroup.org/HDF5/doc/Specs.html`
7. ISO 19123:2005. Geographic information – Schema for coverage geometry and functions. Standard, International Organization for Standardization, Geneva, CH (Aug 2005)

8. ISO 19156:2011. Geographic information – Observations and measurements. Standard, International Organization for Standardization, Geneva, CH (Aug 2011)
9. JSON-LD 1.0. Standard, W3C (Jan 2014), `https://www.w3.org/TR/json-ld/`
10. PROV-O: The PROV Ontology. Standard, W3C (Apr 2013), `https://www.w3.org/TR/prov-o/`
11. QB4ST: RDF Data Cube extensions for spatio-temporal components. Editor's draft, W3C (Dec 2016), `http://w3c.github.io/sdw/qb4st/`
12. The RDF Data Cube Vocabulary. Standard, W3C (Jan 2014), `https://www.w3.org/TR/vocab-data-cube/`
13. Rew, R., Davis, G.: NetCDF: an interface for scientific data access. IEEE Computer Graphics and Applications 10(4), 76–82 (July 1990)
14. Semantic Sensor Network Ontology. Working draft, W3C (May 2016), `https://www.w3.org/TR/vocab-ssn/`
15. WCS 2.0 Interface Standard - Core. Standard 09-110r4, Open Geospatial Consortium (July 2012), `http://www.opengis.net/doc/IS/wcs-core-2.0.1`

A Sample CoverageJSON document

This is a coverage that measures a single parameter (air temperature) on a regular global grid of 1 degree resolution.

```
{
  "type" : "Coverage",
  "domain" : {
    "type": "Domain",
    "domainType": "Grid",
    "axes": {
      "x": { "start": -179.5, "stop": 179.5, "num": 360 },
      "y": { "start": -89.5, "stop": 89.5, "num": 180 },
      "t": { "values": ["2013-01-13T00:00:00Z"] }
    },
    "referencing": [{
      "coordinates": ["x","y"],
      "system": {
        "type": "GeographicCRS",
        "id": "http://www.opengis.net/def/crs/OGC/1.3/CRS84"
      }
    }, {
      "coordinates": ["t"],
      "system": {
        "type": "TemporalRS",
        "calendar": "Gregorian"
      }
    }]
  },
  "parameters" : {
    "TEMP": {
```

```
      "type" : "Parameter",
      "description" : {
        "en": "The air temperature measured in degrees Celsius."
      },
      "unit" : {
        "label": {
          "en": "Degree Celsius"
        },
        "symbol": {
          "value": "Cel",
          "type": "http://www.opengis.net/def/uom/UCUM/"
        }
      },
      "observedProperty" : {
        "id" : "http://vocab.nerc.ac.uk/standard_name/air_temperature/",
        "label" : {
          "en": "Air temperature",
          "de": "Lufttemperatur"
        }
      }
    }
  },
  "ranges" : {
    "TEMP" : "http://example.com/coverages/123/TEMP"
  }
}
```

B Result of converting CoverageJSON document to RDF via JSON-LD

The following document (in the Turtle notation) shows the result of automatically converting the coverage from Appendix A to RDF using the JSON-LD context. Some information has been omitted for brevity and clarity.

```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>
@prefix covjson: <https://covjson.org/def/core#>
@prefix covjsondt: <https://covjson.org/def/domainTypes#>
@prefix ssn: <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn#>
@prefix skos: <http://www.w3.org/2004/02/skos/core#>
@prefix dct: <http://purl.org/dc/terms/>
@prefix qudt: <http://qudt.org/schema/qudt#>
@prefix ignf: <http://data.ign.fr/def/ignf#>
@prefix inspiregloss: <http://inspire.ec.europa.eu/glossary/>

<http://example.com/mycoverage> a covjson:Coverage;
```

```
  covjson:domain [
    a covjson:Domain;
    covjson:domainType covjsondt:Grid;
    covjson:referencing: [
      covjson:referenceSystem: <http://www.opengis.net/def/crs/OGC/1.3/CRS84>
    ], [
      covjson:referenceSystem: [
        a inspiregloss:TemporalReferenceSystem ;
        covjson:calendar: <http://www.opengis.net/def/uom/ISO-8601/0/Gregorian> .
      ]
    ]
  ] ;
  covjson:parameter [
    a covjson:Parameter ;
    dct:description "The air temperature measured in degrees Celsius."@en ;
    qudt:unit [
      qudt:symbol "Cel"^^<http://www.opengis.net/def/uom/UCUM/> ;
      skos:prefLabel "Degree Celsius"@en
    ] ;
    ssn:observedProperty <http://vocab.nerc.ac.uk/standard_name/air_temperature/>
  ] .

<http://www.opengis.net/def/crs/OGC/1.3/CRS84> a ignf:GeodeticCRS .

<http://vocab.nerc.ac.uk/standard_name/air_temperature/>
  skos:prefLabel "Lufttemperatur"@de ;
  skos:prefLabel "Air temperature"@en .
```