# netCDF-LD SKOS: Demonstrating Linked Data Vocabulary Use Within netCDF-Compliant Files

Nicholas Car, Alex Ip, Kelsey Druken



# netCDF-LD SKOS: demonstrating Linked Data vocabulary use within netCDF-compliant files

Nicholas J. Car[1], Alex Ip[1] and Kelsey Druken[2]

[1]Geoscience Australia, Symonston, ACT, Australia, [2]Australian National Computational Infrastructure, Acton, ACT, Australia

{nicholas.car, alex.ip}@ga.gov.au, kelsey.druken@anu.edu.au

**Abstract.** netCDF, the widely-used array-oriented data container file format, has previously been extended in an initiative called netCDF-LD, to include Linked Data metadata elements. In this paper, we build on that initiative with demonstrations of a Simple Knowledge Organization System (SKOS)-aware file format and associated tooling. First, we discuss a very simple way to reference SKOS vocabulary data stored online in netCDF files via Linked Data. Second, we describe our prototype *ncskos* tools, including '*ncskosdump*', which wraps the well-known '*ncdump*' tool used to print out netCDF headers and data. Our tools utilize some of the features of Linked Data and SKOS vocabularies to enhance the metadata of netCDF files by allowing: multilingual metadata label retrieval; alternate term name retrieval; and hierarchical vocabulary relationship navigation. In doing this, *ncskosdump* preserves the *ncdump* practice of writing output in standard CDL (network Common Data Language). For the demonstration of theses formats and tools, we relate how we have included URI links in netCDF files to SKOS concepts within a demonstration vocabulary and how the *ncskos* tools can be used to manage these files in ways that are not possible using only regular netCDF metadata. We also discuss problems we perceived in scaling Linked Data functionality when applying it to large numbers of netCDF files or in multiple file management sessions, and how we have catered for these. Finally, we indicate some future work in the area of more comprehensive Linked Data representation in netCDF files.

**Keywords:** netCDF, netCDF-LD, Linked Data, inference, vocabularies, SKOS

## 1 Introduction

NetCDF files [1] are containers that include both data – usually array-oriented scientific data – and metadata. The intention of the metadata inclusion is that netCDF files should be 'self-describing' meaning the metadata (usually referred to as being in a 'header') describes the rest of the file (the data), as well as an arbitrary number of name/value attributes which together allow for accurate interpretations of the data and thus sensible use of it. Some communities, such as the Climate and Forecasting (CF) Conventions [2], use standardized name/value attributes to enhance metadata understanding between

different parties leading towards interoperability of netCDF data. In most cases, those communities achieve interoperability by constraining the metadata values, potentially reducing its richness.

The Semantic Web [3] is a set of extensions to Web standards that allow for data exchange and, ultimately, knowledge representation. Collections of terms (vocabularies) and knowledge graphs (ontologies) can be codified and, through Linked Data principles [4], be published and accessed. The ability for Linked Data links in any Internet-connected system to give live access to rich vocabulary and ontology information published on Web is potentially very useful for netCDF data. Such links in file metadata allow associations with further, more detailed, metadata, meaning that not all of the metadata relevant to a netCDF file needs to be contained within the netCDF container itself. While this may seem to break with the netCDF aim of the format being self-describing, we believe that conventions that maintain any information externally to netCDF files, such as the CF Convention's relations between terms given in a website hierarchy, have already broken with this (what we believe to be impossible) aim.

Agreed systems of knowledge representation (ontologies) can be used to define the type (classes) of knowledge within a domain. One such ontology is the Simple Knowledge Organization System (SKOS) [5] which has been made to represent items within vocabularies and thesauri. Rather than constraining the metadata values of netCDF files to achieve interoperability, Semantic Web and Linked Data methodologies can be used to constrain the mechanisms for describing and accessing metadata without constraining the content itself. In addition to achieving interoperability via the use of SKOS, since it is a standardized and a well-known ontology, SKOS use also instantly provides netCDF file makers with access to a large number of already published SKOS vocabularies and their content, such as those employed by Geoscience Australia[1] and several versions of the CF Conventions' terms which have recently been published[2].

Sophisticated methods for the inclusion of Linked Data within netCDF files have been demonstrated at a previous ISESS conference called 'netCDF-LD' [6] and an example of netCDF-LD using Linked Data reasoning has also been given recently [7]. This paper is both a 'next step' to those bodies of work and also a step back with respect to the complexity of implementation. We present a simplified mechanism for referencing SKOS data within a netCDF file (Section 2) which, we believe, is easily understood by users of regular netCDF data, and yet which delivers at least some the benefits that a fully-fledged combination of netCDF and Linked Data offer. Due to the simple nature of our encoding mechanism, we can demonstrate a simple wrapper for the well-known *ncdump* command line tool, which we call *ncskosdump*, which prints out netCDF data and metadata. Details of this tool are given in Section 3. In Section 4, we describe a test deployment of SKOS data to netCDF files that references a demonstration SKOS vocabulary that we have built. Vocabularies like this are to be used in place of community-

[1] http://pid.geoscience.gov.au/def/voc/

[2] http://vocab.nerc.ac.uk/collection/P07/current/. The same SKOS vocabulary is republished at http://auscope-services-test.arrc.csiro.au/elda-demo/nerc/resource?_view=skos&uri=http://vocab.nerc.ac.uk/collection/P07/current/

agreed conventions for the classification of those files, using features of the vocabulary, using *ncskos* tools. In Section 5, we discuss how the netCDF combined with the Linked Data and Semantic Web methodologies and *ncskos* tools we have presented here might be made more efficient when used at scale. Finally, in Section 6, we indicate a few areas of future work regarding more comprehensive linked data representation in netCDF files.

## 2 Simple Linked Data encoding

While Yu et al. [5] proposed and Baird et al. [6] have demonstrated sophisticated integration of Linked Data into netCDF files, we have tested simples of integrations only whereby we place Universal Resource Indicator (URI) links [8] to SKOS concepts within attribute metadata in netCDF files. Our reason for taking this very simple approach is to enable the demonstration, and ultimately the adoption, of some limited Linked Data and Semantic Web functionality with the lowest possible barrier to uptake. This link inclusion approach delivers files that very closely resemble 'normal' netCDF files that conform with the netCDF4 specification. Figure 1 shows some basic netCDF metadata taken from a file used by the Unidata community for the demonstration[3] of metadata extraction using the *ncdump* tool.

```
float tos(time, lat, lon) ;
  tos:standard_name = "sea_surface_temperature" ;
  tos:long_name = "Sea Surface Temperature" ;
  tos:units = "K" ;
  tos:cell_methods = "time: mean (interval: 30 minutes)" ;
  tos:_FillValue = 1.e+20f ;
  tos:missing_value = 1.e+20f ;
  tos:original_name = "sosstsst" ;
  tos:original_units = "degC" ;
```

**Fig. 1.** A sample of normal netCDF metadata for the time variable within an example file[2].

```
float tos(time, lat, lon) ;
  tos:skos__concept_uri = "http://pid.geosci-
ence.gov.au/def/voc/netCDF-LD/sea_surface_temperature";
  tos:units = "K" ;
  tos:cell_methods = "time: mean (interval: 30 minutes)" ;
  tos:_FillValue = 1.e+20f ;
  tos:missing_value = 1.e+20f ;
  tos:original_units = "degC" ;
```

[3] http://www.unidata.ucar.edu/software/netcdf/examples/files.html, see table "Sample files following CF conventions"

**Fig. 2.** Similar netCDF metadata to Fig. 1 with key/value pairs removed and a `skos_concept_uri` key/value pair added with the URI of a concept in a demonstration vocabulary[4].

Figure 2 shows this same header metadata with the standard_name key/value pair replaced with a URI link (key: skos_concept_uri) to a segment of a SKOS vocabulary; the entry for the concept of "sea surface temperature". This linking allows a whole set of SKOS data – the entire vocabulary the concept is drawn from – to be associated with the netCDF file containing the link via link look-up ("dereferencing"), rather than just the single textual value of "sea_surface_temperature" which may have further information associated with it which is not discoverable in any standardized way. The link in this example is to data in a demonstration vocabulary that we have built to replicate a small portion of the CF Conventions' [2] terms, with some of our own additions for testing purposes, relating to surface temperature using the SKOS ontology.

## 3 The *ncskosdump* tool

The netCDF specification includes a set of software tools used to manipulate netCDF files [6]. These tools allow people to view netCDF metadata and data, create netCDF files and view visual representations of netCDF's gridded arrays. Some examples of tools, their developers and their usages are given in Table 1.

**Table 1.** Several netCDF tools

| Name | Creators | Purpose |
|---|---|---|
| ncdump[5] | Unidata, the makers of netCDF | Command-line utility converts netCDF data to human-readable text form (CDL or XML) |
| ncgen[6] | Unidata | A program that creates a netCDF dataset fro CDL input |
| ncview[7] | Scripps Institution of Oceanography | A netCDF visual browser |
| The netCDF Operators toolkit[8] | A community of developers | Command-line programs that take netCDF, HDF, and/or DAP files and derive new data, compute statistics, print or otherwise manipulate them |

In order to make our simple SKOS-only deployment of netCDF-LD usable by people without detailed knowledge of Linked Data, we have created a prototype Python[9] utility

[4] Demo vocabulary online at http://pid.geoscience.gov.au/def/voc/netCDF-LD-st-demo
[5] http://www.unidata.ucar.edu/software/netcdf/netcdf-4/newdocs/ncdump-man-1.html
[6] http://www.unidata.ucar.edu/software/netcdf/netcdf-4/newdocs/ncgen-man-1.html
[7] http://meteora.ucsd.edu/~pierce/ncview_home_page.html
[8] http://nco.sourceforge.net/
[9] The Python programming language, v2.7: https://www.python.org/download/releases/2.7/

called *ncskosdump* which wraps the *ncdump* tool (see Table 1) and adds a series of options for the retrieval and display of Linked Data that the tool is able to extract from a netCDF file, including links to vocabulary terms as described in Section 2.

*ncskosdump* finds and dereferences SKOS concept URIs to access vocabulary metadata and uses a series of SPARQL queries [9] on that data to present requested subsets of it to users.

The tool is presented as a Git[10]-based code repository online which is catalogued at http://pid.geoscience.gov.au/dataset/103620 (repository version 1.0). That repository contains comprehensive documentation and test code & data for the tool, including all examples used in this paper. The vocabulary used for testing is online at http://pid.geoscience.gov.au/def/voc/netCDF-LD-eg-ToS (version 1.1) and a copy of its content is stored within the repository in the file examples/tos.ttl.

## 4 A deployment scenario

A series of small netCDF test files (included in the data directory of the code repository) has been created to test and demonstrate the Linked Data functionality of the *ncskos* tools, including *ncskosdump*. One file was created for each valid concept in the test vocabulary, and a "*skos__concept_uri*" variable attribute in the file set to the URI of the Concept. In addition, one file was created with its "*skos__concept_uri*" variable attribute value set to an un-resolveable ("non-dereferenceable", in Linked Data jargon), dummy URI; another with a valid global "*skos__concept_uri*" attribute (as opposed to an attribute of a variable); and yet another with no "*skos__concept_uri*" global or variable attribute value defined for error case testing.

The *ncskosdump* command, like its ancestor, *ncdump*, could be invoked within a script for each netCDF file, and its Common Data Language (CDL) or eXtensible Markup Language (XML) output parsed to infer relationships between concepts. This approach would, however, be extremely inefficient, so Python classes (*ConceptHierarchy* and *NCConceptHierarchy*) have been implemented in order to facilitate efficient programmatic handling of large numbers of files and the resolution of their URIs using the same Linked Data mechanisms implemented for *ncskosdump*.

The script *skos_inferencing_demo.py* inspects each of the specified netCDF files, resolves its URI(s), where possible, and then displays the file paths and variable names grouped hierarchically by Concept as indented lists, as shown in Figure 3. The Concept hierarchies are inferred from the broader/narrower results of the SKOS queries. The Concepts are cached in memory within a session, so each Concept is resolved only once across multiple files, and cached on persistent storage between sessions by using YAML[11] files of concept retrieval results. By default, the concept hierarchies are populated recursively from each linked Concept all the way up to the Top Concept(s) of the vocabulary to provide the full context of the Concepts in the files. It is also possible to recursively build the full Concept hierarchy downward to the narrowest Concepts by specifying the "--narrower" command line option.

[10] Git is a distributed version control system: https://git-scm.com/

[11] YAML is a human-readable data serialization language: https://en.wikipedia.org/wiki/YAML

```
> python skos_inferencing_demo.py --lang=pl C:\data
…
temperatura powierzchni
  C:\data\sst.ltm.1999-2000_skos_surface_temperature.nc:sst
        temperatura powierzchni morza
          C:\data\sst.ltm.1999-2000_skos_sea_surface_temperature.nc:sst
                sea surface skin temperature (English)
                  C:\data\sst.ltm.1999-2000_skos_sea_surface_skin_tempera-
ture.nc:sst
                sea surface subskin temperature (English)
                  C:\data\sst.ltm.1999-2000_skos_sea_surface_subskin_tem-
perature.nc:sst
                temperatura powierzchni morza do kwadratu
                  C:\data\sst.ltm.1999-2000_skos_square_of_sea_surface_tem-
perature.nc:sst

Unresolved URI dummy_uri
  C:\data\sst.ltm.1999-2000_skos_dummy_uri.nc:sst

Uncategorised (Missing URI)
  C:\data\sst.ltm.1999-2000_skos.nc
```

**Fig. 3.** Sample output from *skos_inferencing_demo.py* showing netCDF files and variables in hierarchical groupings with Polish language labels. Note the fallback to English where Polish prefLabels are not defined.

In order to cater for another practical application of SKOS, the script skos_inferencing_demo.py also takes an optional command line argument "--altlabels=<altLabels>", where <altLabels> is a comma-separated list of case-sensitive altLabels (alternative names for a label) for which netCDF files and variables matching a corresponding or narrower Concept are listed. Sample output is shown in Figure 4.

Current command line options for skos_inferencing_demo.py are as follows:

***--verbose*** to enable verbose output

***--lang=<lang_code>*** where <*lang_code*> is a two-character ISO 639-1:200 code for the language in which the results are sought

***--narrower*** to recursively create complete tree of narrower concepts, not just ones resolved directly from URIs

***--altLabels=<altLabel_list>*** where <*altLabel_list*> is a comma-separated list of altLabels to match in order to list their associated datasets

***--retries=<max_retries>*** where <*max_retries*> is the maximum number of retries to attempt for unresolved URIs. Default retries = 0

***--delay=<retry_delay_seconds>*** where <*retry_delay_seconds*> is the number of seconds to wait before each retry. Default delay = 2s

***--refresh*** to discard current file cache and repopulate the cache from scratch

```
> python skos_inferencing_demo.py --lang=pl –altlabels=SST,sst C:\data
…
altLabel matches

Concepts and datasets with altLabel "SST":
temperatura powierzchni morza
  C:\Users\Alex\git\ncskosdump\data\sst.ltm.1999-2000_skos_sea_surface_tem-
perature.nc:sst
        Narrower Concepts:
        sea surface skin temperature (English)
          C:\Users\Alex\git\ncskosdump\data\sst.ltm.1999-2000_skos_sea_sur-
face_skin_temperature.nc:sst
        sea surface subskin temperature (English)
          C:\Users\Alex\git\ncskosdump\data\sst.ltm.1999-2000_skos_sea_sur-
face_subskin_temperature.nc:sst
        temperatura powierzchni morza do kwadratu
          C:\Users\Alex\git\ncskosdump\data\sst.ltm.1999-
2000_skos_square_of_sea_surface_temperature.nc:sst

No concepts found with altLabel "sst"
```

**Fig. 4.** Sample output from *skos_inferencing_demo.py* showing search results using two altLabels “SST” & “sst”. NetCDF files which match the corresponding concept, or its narrower concepts, are shown in hierarchical groupings with Polish language labels (where available).

## 5 Consideration of scaling and retrieval strategies

The *ncskos* Python classes used in the current version of *skos_inferencing_demo.py* utilize in-memory caching of Concepts retrieved via SKOS queries within a single session and caches whole Concept hierarchies and their information on disk in YAML for use between equivalent sessions (i.e. sessions with the same SKOS options). This means that for most operations, the *ncskos* Python classes need only resolve (dereference) each Concept URI once which, for large numbers of files with the same URIs, greatly reduces network overheads and thus improves program performance. As with any client-side caching system, care needs to be taken to ensure that cached data is kept consistent with the point of truth (i.e. the source vocabulary online), hence the command matching and the ability for users to refresh the file cache. The current version of *skos_inferencing_demo.py* also resolves concept URIs recursively as required.

## 6 Future Work

Resolution strategy changes are likely with a better understanding of the tool’s usage patterns, as indicated above. An alternative approach for efficient URI resolution might be to read all URIs within netCDF files and then resolve the multiple URIs in bulk using a much smaller number of queries. Where a complete concept hierarchy is required and the source(s) is/are resolvable, it may also be better to retrieve the entire

concept hierarchy from each relevant vocabulary before matching individual URIs. The decision as to which approach is best for given scenarios cannot be determined until a larger number of community uses of this tool are available for analysis.

Additionally, extensions to the tool's ability to handle more forms of Linked Data encoded in netCDF files will surely be considered. Already the potential for handling multiple URIs per global or variable attribute (as a comma-separated list) has been implemented and tested during development, but was deemed to be problematic at this stage because of the potential for inconsistent or nonsensical user-defined combinations of URIs. This remains an area open to further investigation, since it may be very useful to be able to reference multiple, independent concept URIs if appropriate validation can be undertaken.

Current work being undertaken by the authors and members of a larger "netCDF-LD working group" are looking in to methods of full Linked Data representation within netCDF files, as indicated in [7]. Results from this work make prompt changes to SKOS Concept URI representation within netCDF files, perhaps through the use of URI and URI prefix (the "*skos__concept_uri*" key) aliasing that is more akin to Linked Data conventions.

## 7 Conclusions

Metadata "bloat" is a very real issue for many of today's scientific datasets, and it can be difficult to maintain consistency where metadata is replicated within data files. For example, complex hierarchical relationships are difficult to represent in metadata encapsulated within netCDF files and where such information is stored externally to the files, such as the CF Conventions community has done, that information may not be either easily discoverable or machine-readable. Linked Data provides a mechanism which can address these issues.

The simple Linked Data functionality demonstrated in our *ncskos* implementation can reference much richer metadata for netCDF files in standardized ways, including alternate language representations and the complete broader/narrower context of concepts as required. The attribute containing URIs to SKOS Concepts can coexist with existing netCDF metadata conventions such as the Climate and Forecasting (CF) [2] or Attribute Convention for Data Discovery (ACDD) [10] so that backwards-compatibility with 'normal' netCDF files and file use can be maintained. In addition, at least some of the well-known conventions (CF) have already been represented in SKOS vocabularies thus we expect that many of those initiatives will come to publish their term lists in ways compatible with this approach.

Our code has been written expressly to handle large numbers of netCDF files with multiple Linked Data concepts, and we are looking to move to real-world implementation in the very near future.

## 8 Acknowledgements

The authors thank the members of the netCDF-LD Working Group for their stimulation of work in this area and their thoughts around Linked Data representation in netCDF.

This paper is published with the permission of the CEO, Geoscience Australia.

## 9 References

1. UCAR, "Network Common Data Form (NetCDF)". Web Page, online at http://www.unidata.ucar.edu/software/netcdf/, retrieved 2016-09-29.
2. Eaton, B., Gregory, J., Drach, B., Taylor, K., Hankin, S.: NetCDF Climate and Forecast (CF) Metadata Conventions (2011). Online at http://cfconventions.org/cf-conventions/v1.6.0/cf-conventions.html, retrieved 2017-01-09.
3. Berners-Lee, Tim; James Hendler; Ora Lassila (May 17, 2001). "The Semantic Web". *Scientific American Magazine*. Online at http://www.scientificamerican.com/article/the-semantic-web/, retrieved 2016-09-29.
4. Tim Berners-Lee. "Linked Data". *Design Issues*. W3C. Online at https://www.w3.org/DesignIssues/LinkedData.html, retrieved 2016-09-29.
5. Alistair Miles & Sean Bechhofer (eds.), SKOS Simple Knowledge Organization System Reference. W3C Recommendation 18 August 2009. Online at https://www.w3.org/TR/skos-reference/, retrieved 2016-09-29.
6. J. Yu, N. J. Car, A. Leadbetter, B. A. Simons, and S. J. D. Cox, "Towards Linked Data Conventions for Delivery of Environmental Data Using netCDF," in *Environmental Software Systems. Infrastructures, Services and Applications: 11th IFIP WG 5.11 International Symposium, ISESS 2015, Melbourne, VIC, Australia, March 25-27, 2015. Proceedings*, R. Denzer, R. M. Argent, G. Schimak, and J. Hrebíček, Eds. Cham: Springer International Publishing, 2015, pp. 102–112. doi: 10.1007/978-3-319-15994-2_9.
7. Jim Biard, Jonathan Yu, Mark Hedley, Simon J D Cox, Adam Leadbetter, Nicholas John Car, Aaron Sweeney, Kelsey A Druken, Stefano Nativi, Ethan Davis "Linking netCDF Data with the Semantic Web - Enhancing Data Discovery Across Domain" in AGU Fall Meeting *Advancing netCDF-CF for the Geoscience Community*, San Francisco, 2015.
8. Joint W3C/IETF URI Planning Interest Group, "URIs, URLs, and URNs: Clarifications and Recommendations 1.0". Report from the joint W3C/IETF URI Planning Interest Group, W3C Note 21 September 2001. Online at https://www.w3.org/TR/uri-clarification/ retrieved 2016-09-29.
9. Eric Prud'hommeaux & Andy Seaborne (eds.), "SPARQL Query Language for RDF". W3C Recommendation 15 January 2008. Online at https://www.w3.org/TR/rdf-sparql-query/, retrieved 2016-09-29.
10. ESIP Federation, "Category:Attribute Conventions Dataset Discovery". Wiki web page, latest revision Jan. 2015. Online at http://wiki.esipfed.org/index.php?title=Category:Attribute_Conventions_Dataset_Discovery, retrieved 2017-01-05.