SISSVoc: A Linked Data API for SKOS vocabularies

Simon J D Cox[a*], Jonathan Yu[a] and Terry Rankine[b]
[a] *CSIRO Land and Water, PO Box 56, Highett, Vic. 3190 Australia*
[b] *CSIRO Earth Science and Resource Engineering, PO Box 1130, Bentley WA, 6102 Australia*
{first.last}@csiro.au

Abstract. The Spatial Information Services Stack Vocabulary Service (SISSVoc) is a Linked Data API for publishing vocabularies. SISSVoc provides a RESTful interface via a set of URI patterns that are aligned with SKOS. This provides a standard web interface to any controlled vocabulary structured using, or decorated with, SKOS classes and properties. It can be consumed via web clients as human-readable resources (such as HTML) and by client applications through machine-readable resources (such as RDF, JSON, and XML). SISSVoc is implemented using a Linked Data API façade over a SPARQL endpoint. The use of the Linked Data Approaches streamlines the configuration of content negotiation, styling, query construction and dispatching. SISSVoc is being used in a number of projects, mainly in the environmental sciences, where controlled vocabularies are used to support cross-domain and interdisciplinary interoperability. The SISSVoc standard interface makes it possible for the development of common client applications such as search clients and validation clients.

Keywords: Vocabulary, SKOS, API, Linked data

1. Introduction

Controlled vocabularies are a key element of many classification systems. They are typically published by specific organisations, domains, or communities of practice. The web has encouraged and enabled consolidation of vocabulary use, such that common vocabularies are now more likely to be maintained and published at a community level than only within an agency or project team, thus improving interoperability of scientific datasets. Examples include chemical entities [11,17], bio-medical terminology [27,36,37], environmental science topic or subject headings [12,21,23] and geological classifications (see compilation at [22]). Vocabularies such as EuroVoc [38] and the International Chronostratigraphic Chart [6] have very well-defined governance and authority, i.e. the Publications Office of the European Union, and the International Commission for Stratigraphy, respectively.

While many vocabularies are openly available on the web, they are formalized and published in a variety of generally incompatible ways, including databases and spreadsheets, text documents, page and image formats. Some of the most fundamental vocabularies are made available on the web by their official custodian only as browser pages or PDFs for download (e.g SI units of measure[1], geologic timescale[2]).

The emergence of *Semantic Web* technologies has provided some powerful tools for formalizing definitions, vocabularies, and ontologies, in forms that also support reasoning and inferencing. In this context, the Simple Knowledge Organization System (SKOS) [20][1] was designed to allow easy formalization of existing multilingual vocabularies that have flat or hierarchical structures, to smooth the transition towards the richer logic-based tools from ontology modelling.

SKOS provides a standard vocabulary for representing thesauri, classifications, taxonomies and controlled vocabularies, using RDF. SKOS has a simple model with few key constructs, focussing on labelling and basic hierarchies. While it lacks the expressivity and rigour of languages such as OWL, its simplicity allows a broad range of vocabularies and classifiers to be ported from a diverse set of formats to RDF, promoting ease of sharing and cross-linking between vocabularies. Many existing vocabularies

[1] http://www.bipm.org/en/si/base_units/, http://www.bipm.org/en/si/si_brochure/
[2] http://stratigraphy.org/index.php/ics-chart-timescale

have being ported to SKOS [19] including large vocabularies such as AGROVOC [25] and the Library of Congress Subject Headers (LCSH) [31]. SKOS is now one of the most commonly used vocabularies for structured data on the web [19].

Despite successes in the use of SKOS for *encoding* vocabularies, current standards only provide very low-level *interfaces* to vocabulary data. Many vocabularies are published as an RDF document for download. However, if the vocabulary is large then the download will be commensurately large, and in many cases the user only wants to retrieve a single vocabulary term or select a few terms, so the download option requires processing on the client side. Alternatively, access to many vocabularies is provided at a SPARQL endpoint. SPARQL [16,24] is the generic RDF query language. While this is very powerful, it is a low-level language similar to the relational database query language SQL. Some SKOS vocabularies are published via various HTTP interfaces. However, each implementation use different protocols and support for a range of features such as content-negotiation varies such as GEMET [39] (REST interface), and NERC (BODC) Data Grid's Vocabulary Server [18,40] (SOAP interface). In some cases, either one or both of human-readable formats and machine-readable formats is not available. Thus, discovery and access across vocabulary endpoints becomes challenging and ad-hoc.

There is a clear opportunity here, to design an API to match the SKOS vocabulary, taking advantage of the fact that much modern vocabulary content is structured using SKOS classes and predicates. This API can then be used as the basis for various higher level vocabulary applications.

Linked Data has been proposed as a means of publishing and interlinking structured data on the web. Linked Data proposes the use of RDF for describing structured data and allows relationships and links between resources to be defined [4]. This allows both human-readable and machine-readable content/interaction to access data resources and their descriptive metadata using existing web technologies simply by dereferencing HTTP URIs. A number of SKOS vocabulary services are available that utilise Linked Data approaches such as Semantic Technologies for Archaeological Resources (STAR) Project's semantic terminology services, Library of Congress Authorities and Vocabularies service, and the Coastal and Marine Spatial Planning Vocabularies (CMSPV) SKOS API [41]. However, as with the above examples, each of these services have different interfaces to access the content, requiring users to have bespoke means of querying for vocabulary resources. Technologies such as Pubby [9], D2R server [3] and Epimorphics Linked Data API Implementation (ELDA) [13] are available for publishing RDF resources as Linked Data. Nevertheless, the gap here is a standard interface for access to SKOS vocabulary resources. The fundamental issue is that RESTful approaches rely only on URIs, HTTP, and content-types [14,26], yet SKOS is not recognised as a 'content-type' in this context.

In this paper, we describe a standard interface called SISSVoc through which SKOS vocabularies can be provided to web users. SISSVoc provides a level of abstraction for the end users corresponding to the SKOS content model, so SISSVoc can be used much like any other web application without specific knowledge of the underlying technologies and semantic web languages used, such as SPARQL endpoints and queries, SKOS and RDF. A human interface in the form of HTML web pages and forms may be made available via HTTP content negotiation. But primarily SISSVoc also for machine-to-machine use, so that data providers can use HTTP links to vocabularies, data applications can be configured with standard terminology, and data clients can retrieve definitions or verify the existence of items claimed to be in particular vocabularies. SISSVoc v1 and v3 have been introduced previously [7,15]. In this paper, we present the SISSVoc v3 design in detail and also describe the current SISSVoc implementation, and evaluate it based on a discussion of its use in the environmental domains and some client applications.

2. SISSVoc design

SISSVoc provides a Linked Data API for publishing SKOS vocabularies. It is designed with a HTTP-based interface that is aligned with RESTful web services [26] and Linked Data [2,4] principles. Standard operations are defined as a set of URI patterns. These URI patterns are aligned with the SKOS vocabulary, to facilitate discovery and access to SKOS vocabulary resources.

The SISSVoc API is normally implemented using a Linked Data API [42] façade over a vocabulary exposed at a SPARQL endpoint. The Linked Data API streamlines the configuration of content negotiation, styling, query construction and dispatching. SISSVoc presents a standard web interface to any controlled vocabulary that is structured using, or at least decorated with, SKOS classes and properties.

Vocabulary content published through SISSVoc may be consumed via web clients as human-readable resources (such as HTML) and client applications with machine-readable resources (such as RDF, JSON, and XML).

The standard interface provided by SISSVoc makes possible the development of common vocabulary applications such as search clients and validation clients. It also aids User Interface (UI) development allowing listboxes and other UI widgets to be populated via HTTP requests to SISSVoc endpoints.

SISSVoc is a key component of the Spatial Information Services Stack (SISS) developed by CSIRO through the AuScope project [34]. Vocabularies formalized in SKOS are accessed via SISSVoc to support discovery of related geodata within the Auscope Portal. SISS is currently being applied more broadly in other environmental information projects.

2.1. SISSVoc API

SISSVoc is designed to deliver descriptions of things as known to the SISSVoc instance, where the URI for the thing is *not* necessarily related to the SISSVoc Service URI. Tables 1-5 show the details of the various parameters in this pattern, and the corresponding SPARQL queries. The general SISSVoc URI pattern is:

```
http://example.org/sissvoc/{type}[/{relation}]
   [?{selection-parameters}[&view-parameters]]
```

A key SISSvoc pattern is the *resource description pattern* (Table 1), in which the description of a resource whose URI is known is obtained using

```
http://example.org/sissvoc/resource
                  ?uri={resourceURI}
```

i.e. type== “resource”,
selection-parameters== “uri={resourceURI}”.

A basic set of SISSVoc URI patterns provide interfaces to query lists of resources of the basic SKOS types. Table 2 lists URI Patterns for querying the set of SKOS ConceptScheme, Collection and Concept respectively.

Another set of SISSVoc URI Patterns provide filtering and selection operations to specific SKOS Concepts. Table 3 lists URI patterns for obtaining a list of concepts based on partial or exact matches on text in labels (rdfs:label, skos:prefLabel, skos:altLabel) for a given vocabulary.

The final set of SISSVoc URI Patterns obtain a list of concepts that are related to a selected concept through the predicates defined in SKOS for structuring vocabularies, i.e. the broader /narrower properties. Table 4 lists URI patterns for broader /broaderTransitive /narrower /narrowerTransitive related to a specific SKOS Concept, denoted by its URI. Table 5 lists URI patterns for obtaining a list of concepts that are broader /broaderTransitive /narrower /narrowerTransitive than SKOS Concepts discovered by the text searches.

SISSVoc is currently specified for HTTP GET operations only.

Table 1

SISSVoc URI Pattern for Resource description

ID	URI pattern	Description	SPARQL
1	/resource?uri={URI}	Resource description identified by URI (not limited to any specific type).	DESCRIBE {URI}

Table 2

SISSVoc URI Patterns for SKOS Concept, ConceptScheme and Collection

ID	URI pattern	Description	SPARQL
2	/conceptscheme	List of all concept schemes	SELECT ?item WHERE { ?item a skos:ConceptScheme }
3	/collection	List of all concept collections	SELECT ?item WHERE {

			?item a ?type . FILTER (?type = skos:Collection || ?type = skos:OrderedCollection)}
4	/concept	List of all concepts	SELECT ?item WHERE { ?item a skos:Concept }

Table 3

SISSVoc URI Patterns for SKOS Concept discovery by label

ID	URI pattern	Description	SPARQL
5	/concept?anylabel={text}	List of concepts where a label matches text	SELECT ?item WHERE { ?item a skos:Concept . ?item ?label ?l . FILTER (?label = skos:prefLabel || ?label = skos:altLabel) FILTER (?l = {text}@en) }
6	/concept?labelcontains={text}	List of concepts where a label contains text	SELECT ?item WHERE { ?item a skos:Concept . ?item ?label ?l . FILTER (?label = skos:prefLabel || ?label = skos:altLabel) FILTER regex(str(?l) , {text} , 'i') }

Table 4

SISSVoc URI patterns for SKOS Concept broader and narrower by URI

ID	URI pattern	Description	SPARQL
7	/concept/broader?uri={URI}	List of concepts skos:broader than the concept identified by URI	SELECT ?item WHERE { ?item a skos:Concept . {URI} skos:broader ?item }
8	/concept/narrower?uri={URI}	List of concepts skos:narrower than concept identified by URI	SELECT ?item WHERE { ?item a skos:Concept . {URI} skos:narrower ?item }
9	/concept/broaderTransitive?uri={URI}	List of concepts skos:broaderTransitive than concept identified by URI	SELECT ?item WHERE { ?item a skos:Concept . {URI} skos:broaderTransitive ?item }
10	/concept/narrowerTransitive?uri={URI}	List of concepts skos:narrowerTransitive than concept identified by URI	SELECT ?item WHERE { ?item a skos:Concept . {URI} skos:narrowerTransitive ?item }

Table 5

SISSVoc URI pattern for SKOS Concept discovery broader/narrower by label

ID	URI pattern	Description	SPARQL

<table>
<tr><td>11</td><td>/concept/broader?anylabel={text}</td><td>List of concepts skos:broader than a concept with a label that matches text</td><td>SELECT ?item
WHERE {
?item a skos:Concept .
?i0 skos:broader ?item .
?i0 ?label ?l .
FILTER (?label = rdfs:label
|| ?label = skos:prefLabel
|| ?label = skos:altLabel)
FILTER (?l = {text}@en)
}</td></tr>
<tr><td>12</td><td>/concept/narrower?anylabel={text}</td><td>List of concepts skos:narrower than a concept with a label that matches text</td><td>SELECT ?item
WHERE {
?item a skos:Concept .
?i0 skos:narrower ?item .
?i0 ?label ?l .
FILTER (?label = rdfs:label
|| ?label = skos:prefLabel
|| ?label = skos:altLabel)
FILTER (?l = {text}@en)
}</td></tr>
<tr><td>13</td><td>/concept/broaderTransitive?anylabel={text}</td><td>List of concepts skos:broaderTransitive than a concept with a label that matches text</td><td>SELECT ?item
WHERE {
?item a skos:Concept .
?i0 skos:broaderTransitive ?item .
?i0 ?label ?l .
FILTER (?label = rdfs:label
|| ?label = skos:prefLabel
|| ?label = skos:altLabel)
FILTER (?l = {text}@en)
}</td></tr>
<tr><td>14</td><td>/concept/narrowerTransitive?anylabel={text}</td><td>List of concepts skos:narrowerTransitive than a concept with a label that matches text</td><td>SELECT ?item
WHERE {
?item a skos:Concept .
?i0 skos:narrowerTransitive ?item .
?i0 ?label ?l .
FILTER (?label = rdfs:label
|| ?label = skos:prefLabel
|| ?label = skos:altLabel)
FILTER (?l = {text}@en)
}</td></tr>
</table>

2.2. Implementing SISSVoc

The SISSVoc API is defined as a set of URI patterns, and corresponding SPARQL queries, which align with the SKOS model to allow discovery and access to vocabulary resources. Aside from the use of HTTP, the API is implementation-neutral allowing for the choice of tools and technologies to be independent of the interfaces.

We have developed a SISSVoc implementation[3] using ELDA [13], an open source implementation of the Linked Data API [42]. ELDA allows configuration of HTTP endpoints corresponding to SPARQL queries for a given RDF triple store (shown in Figure 1). In this implementation, the RDF store/SPARQL endpoint is independent of the SISSVoc deployment, so it is not necessary for them to be co-located. This provides a separation of concerns with regards to where the vocabularies are persisted and maintained, and where the discovery and access interface is deployed.

[3] The CSIRO implementation is documented at https://www.seegrid.csiro.au/wiki/Siss/SISSvoc3Overview. Also see https://github.com/jyucsiro/sissvoc-runner for tool to install locally for testing ELDA configurations.

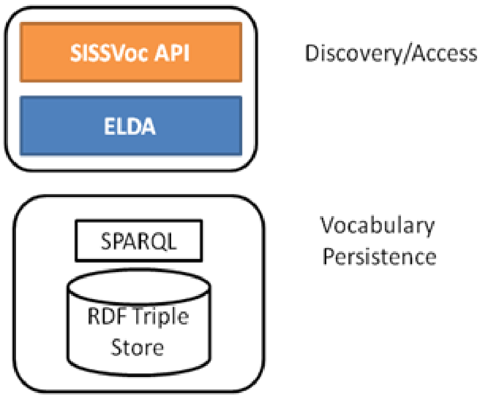

Fig. 1 - SISSVoc Implementation using ELDA

A SISSVoc can be deployed to provide an interface to any SKOS vocabulary that is already published at a SPARQL endpoint. Figure 2 shows an example of this where a SISSVoc deployed at CSIRO (http://auscope-services-test.arrc.csiro.au/elda-demo/nerc/collection) provides a standard interface to the NERC Vocabulary Service (http://vocab.nerc.ac.uk/sparql/).

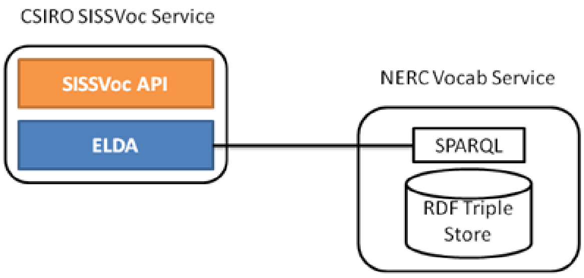

Fig. 2 - Example of a SISSVoc deployment to externally governed vocabulary service

2.3. Deploying SISSVoc to meet use cases

In order to satisfy most vocabulary users, four different interfaces to vocabularies need to be in place:

1. For each **item** in the vocabulary, HTTP GET {URI} should resolve to a description of the item. This is suitable for direct reference to vocabulary items, and in-line links within datasets.

   http://environment.data.gov.au/water/quality/def/object/nitrogen

2. A **SISSVoc interface** to the vocabulary, supporting queries on properties of the vocabulary items, with various options for how the result is formatted and what is included. This is for general users who want to explore a vocabulary without having to know RDF or SPARQL.

   query for all concepts:
   http://sissvoc.ereefs.info/sissvoc/ereefs/concept

   query for concepts narrower than ones with the label "nitrogen":
   http://sissvoc.ereefs.info/sissvoc/ereefs/concept/broader?anylabel=nitrogen

3. A **SPARQL endpoint,** for access to a vocabulary through the SPARQL query language.

   http://sissvoc.ereefs.info/ereefs/sparql

4. The vocabulary bundled as a **single document** (file), delivered from the "Ontology URI". This is for users and services who wish to harvest the whole vocabulary in one transaction.

   http://sissvoc.ereefs.info/vocab/ereefs/wq

The examples here are all from a single instantiation of SISSvoc, but the interfaces are distinct from the user’s point of view. However, each interface uses the next one down for its configuration or real-time operation. Figure 3 shows a typical deployment. A vocabulary maintained in an RDF document is loaded into an RDF triple-store, which presents a SPARQL endpoint that is used by an ELDA instance, which is configured to present a SISSVoc interface. The server(s) for the URIs for items in the vocabulary may be configured so that requests are redirected to the resource-description URI hosted by this SISSvoc, so a HTTP GET will retrieve the graph describing the concept, collection or concept-scheme served by the SISSVoc, and they will stay in the same SISSVoc service when they follow a link to another item in the same vocabulary within the result graph.

2.4. SISSVoc Deployments

SISSVoc is being used in a number of projects, mainly in the earth and environmental sciences, where controlled vocabularies are used to support cross-domain and interdisciplinary interoperability. Examples of vocabularies currently being served using SISSVoc are listed in Table 6.

Table 6

Some examples of SISSVoc deployments

OGC Definitions	http://www.opengis.net/def/
Geological Timescale	http://resource.geosciml.org/classifier/ics/ischart/
Environmental monitoring definitions	http://environment.data.gov.au/water/quality/def/property/ http://environment.data.gov.au/water/quality/def/object/ http://environment.data.gov.au/water/quality/def/unit/
Water and energy supply and consumption (WESC) definitions	http://wescml.org/sissvoc/vocab/collection
ANZSRC Socio-Economic Objective	http://researchdata.ands.org.au:8080/vocab/api/anzsrc-seo/concepts
NERC Vocabularies **(no URI redirection, so links return to the NVS)**	http://vocab.nerc.ac.uk/ via http://auscope-services-test.arrc.csiro.au/elda-demo/nerc/collection

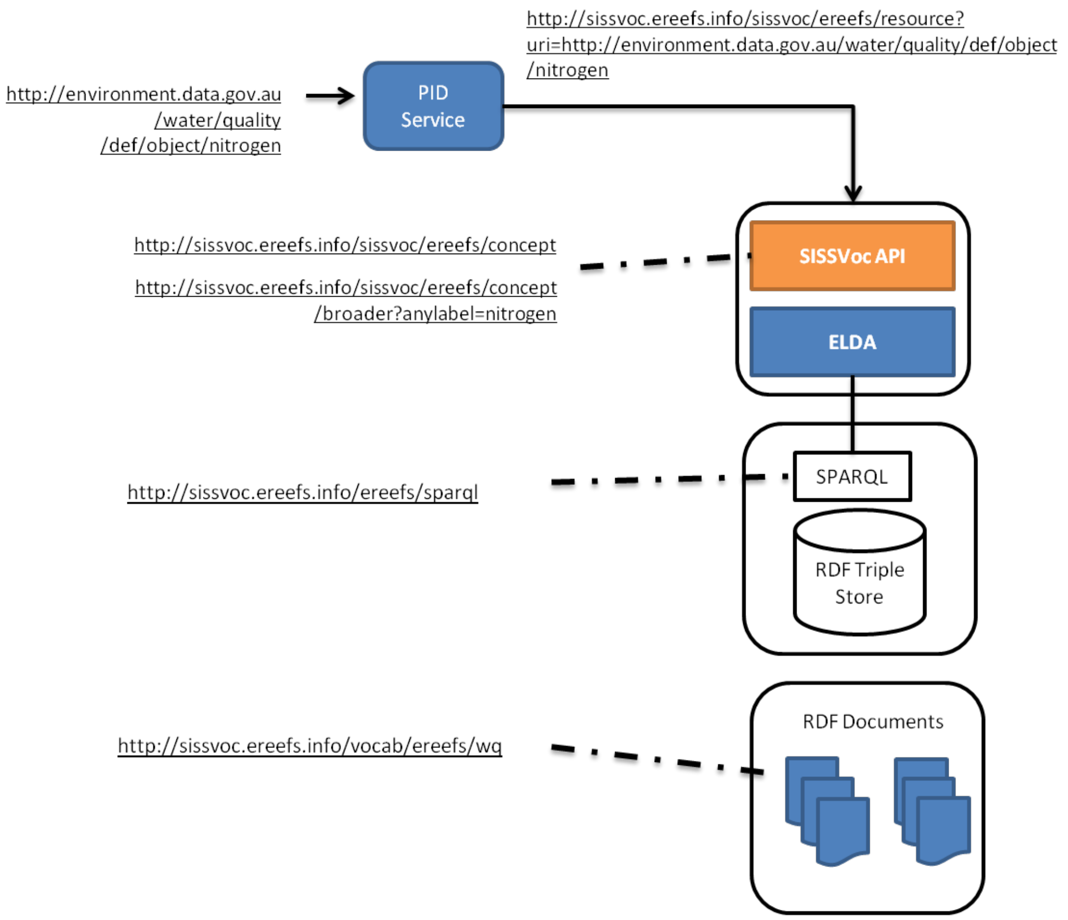

Fig. 3 - SISSVoc Deployment complemented by PID service and a web service hosting RDF documents

3. Evaluation

3.1. URI Patterns

The URI pattern for resource descriptions is

http://example.org/sissvoc/resource?uri={resourceURI}

This may be read as '*what http://example.org/sissvoc/ knows about {resourceURI}*'. It is thus a very explicit implementation of 'Cool URIs for the semantic web' [28]. The pattern can be interpreted in terms of the distinction between *information resources* and *non-information resources*, where in this case

- a 'concept', denoted by {resourceURI}, is understood as an abstract (non-information) resource
- the RDF graph, denoted by the resource description pattern, is a corresponding information resource.

A number of Cool URI patterns have been proposed that distinguish between resources and their descriptions [5,10,28,29]. These combine special tokens in the URI with HTTP parameters and response codes to indicate to the user how to understand the resource. Two are directly comparable with the SISSVoc resource description pattern, in that they have distinct but related URIs for the non-information resource or concept, and for a description of it:

The Cool URIs for the Semantic Web pattern [10,28,29]:

- http://example.org/id/{id} denotes a non-information resource
- http://example.org/doc/{id} a corresponding description or information resource

In DBPedia [5]:

- http://dbpedia.org/resource/{id} denotes a concept, a non-information resource
- http://dbpedia.org/data/{id} an rdf graph describing the concept, an information resource
- http://dbpedia.org/page/{id} an html page describing the concept, an information resource

The SISSVoc pattern:

- http://example.net/{name} denotes a resource (optionally a non-information resource)
- http://example.org/sissvoc/resource?uri=http://example.net/{name} a description or information resource (format selected through content-negotiation)

Note that the resource description pattern (Table 1) does not use the SKOS vocabulary. The properties included in a resource description are those provided by the SPARQL DESCRIBE operation, which is typically an approximation of the Concise Bounded Description [30]. Thus, while the list endpoints described in Tables 2-5 use SKOS predicates, these are intended to lead the user to a description of a selected concept, within which the SKOS elements may be a minor aspect or 'decoration' of a more specific ontology. In the latter context, the SISSVoc SKOS API is a 'generic' access point and may be supplemented by more specific interfaces relevant to a specialized vocabulary. For example, an RDF representation of the 2013 version of the geologic timescale is identified as

http://resource.geosciml.org/classifier/ics/ischart/2013

and delivered by a SISSVoc service in the form of a graph that mixes SKOS predicates with predicates from an ontology designed for the geological timescale [8].

3.2. HTTP operations and REST behaviour

The SISSVoc API is currently only specified for HTTP GET operations. Hence, SISSVoc is not a full RESTful API, as it does not support HTTP operations for update and deletion [14,26]. SISSvoc is intended to be a lightweight search and retrieval SKOS API. Managing vocabulary content is a more challenging task, which involves not only concept descriptions but also all the relationships within and between vocabularies. Maintaining the integrity of these in the face of fine-grained update operations is a significant task. Vocabulary content may be maintained using RDF editors (such as Protégé [4] or TopBraid Composer[5]), which ensure consistency of relationships between resources is maintained, and then generate RDF *documents* to transfer vocabulary content from the maintenance to publication environment, as outlined above. If a web-based vocabulary maintenance environment is required, then tools

[4] http://protege.stanford.edu/

[5] http://www.topquadrant.com/tools/modeling-topbraid-composer-standard-edition/

like TopQuadrant's Enterprise Vocabulary Net[6], and the PoolParty Thesaurus Server[7] are available.

3.3. Applications

The standard interface provided by SISSVoc supports a range of applications.

3.3.1. Water Data Transfer Format validation service

The Water Data Transfer Format (WDTF) is an XML-based exchange standard that was developed for transfer and ingestion of national water data into the Australian Bureau of Meteorology's information systems from over 200 data providers. For quality control, a validation service was implemented combining two standard schema languages to provide structure and content validation [35]. The WDTF validation service (shown below in the blue box of Figure 4) is part of a distributed service-oriented framework for validation which is enhanced with vocabulary checking using the vocabulary service. The service uses a XML schema validation component for structure validation. Schematron is then used to perform content validation such as co-constraint checking and vocabulary checking. Vocabulary checking is achieved by queries to a SISSVoc service hosting WDTF SKOS vocabularies from the Schematron validation process.

3.3.2. SISSVoc Search

The SISSVoc Search tool provides a simple query for vocabulary entries, using domain-specific terms and keywords, built on top of the SISSVoc APIs. SISSVoc Search provides

- a web-based search interface to support search via HTML form interface (http://sissvoc.ereefs.info/search),
- HTTP GET requests, with the query string and SISSVoc endpoint embedded in the URI (e.g. http://sissvoc.ereefs.info/search?q=water&endpoint=http://wescml.org/sissvoc/vocab).

SISSVoc Search allows users to search over one given SISSVoc endpoint with the ability to switch between endpoints. A screenshot of the SISSVoc search interface is shown in Figure 5.

[6] http://www.topquadrant.com/products/topbraid-enterprise-vocabulary-net/

[7] http://www.poolparty.biz/portfolio-item/poolparty-thesaurus-server/

The SISSVoc Search tool provides an interface over the SISSVoc deployments. It allows users to search for vocabulary terms without knowing the specific deployment details other than the deployment URL. This accommodates both a vocabulary interface and search tool independent of the vocabulary service host organisation.

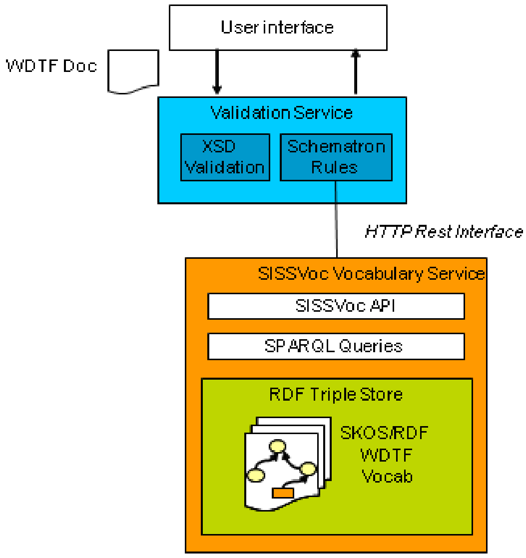

Figure 4 - WDTF Validation Service Leveraging SISSVoc Vocabulary Service

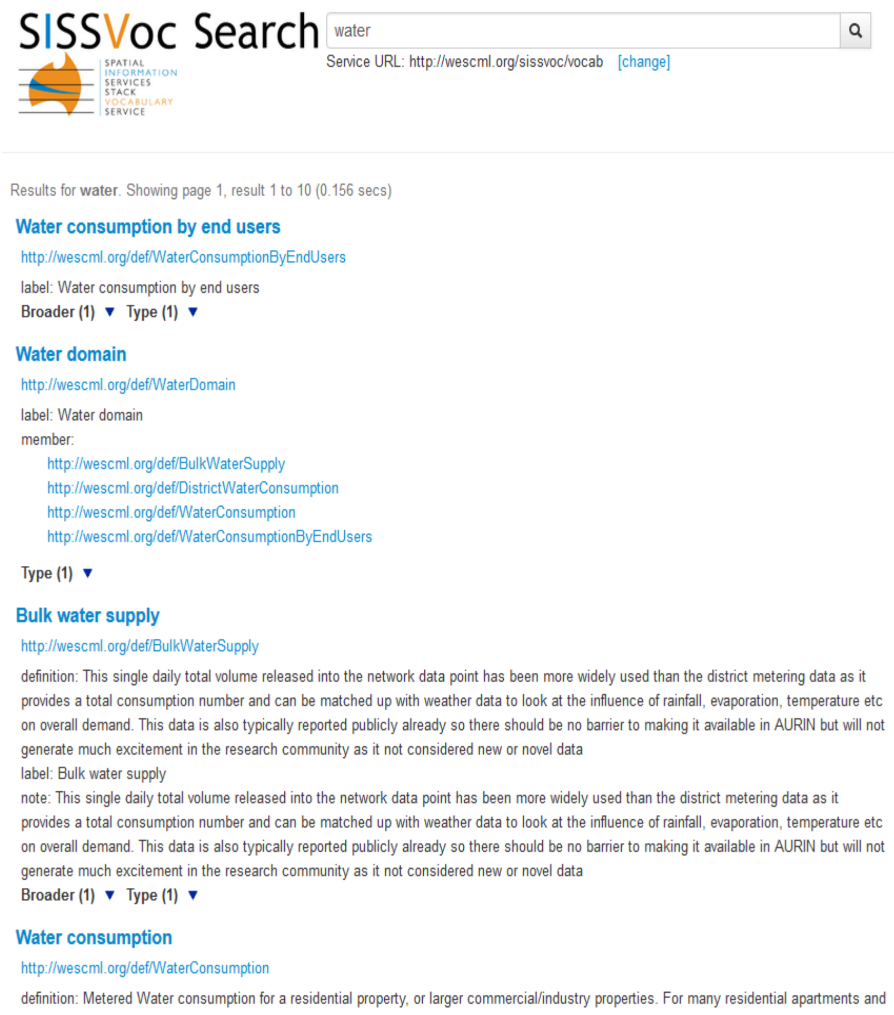

Figure 5 - Screenshot of the SISSVoc search tool

3.4. Related Work

3.4.1. ONKI

ONKI SKOS Server [32] has been developed with similar premises as SISSVoc, to provide interfaces that align with the SKOS resource model. It provides a SOAP Web Service and AJAX interfaces for querying SKOS concepts by label matching, obtaining labels for a given URI[8]. The ONKI SKOS web service interface limits its use to SOAP clients and handling responses in JSON via their AJAX HTTP interface.

ONKI also provides web developers a tool to generate HTML/Javascript code for embedding concept selection via a form based widget. ONKI is implemented in Java and indexes the vocabularies via the Lucene text search engine to support operations such as query expansion. However, these are primarily for human-readable interfaces rather than both human- and machine readable interfaces. This limits its flexibility to accommodate leveraging with other tools and building client applications.

ONKI has focused on providing a suite of tools and interfaces in its implementation, including providing text retrieval functions and support for developers. In contrast, the SISSVoc has focused on

[8] http://onki.fi/api/v2/http

defining APIs and design principles and implementing with current best practices. Both ONKI and SISSVoc have interfaces which align with the SKOS meta-model.

3.4.2. Normalised Ontology Repository (NOR)

NOR [33] proposes two ideas: a normalised presentation for ontology concepts and a simple API for accessing the ontology repository. The former uses SKOS as the normalisation language. The latter specifies a concept lookup method via HTTP GET:

`concept?uri=[concept identifier]`

which is similar to the SISSVoc resource URI pattern, except limited to SKOS concepts.

NOR also features a search method via HTTP GET

`search?q=[query]&l=[language]`

which is similar to the SISSVoc Search HTTP search interface.

The NOR approach is thus similar to SISSVoc, but the set of API operations is significantly smaller than SISSVoc.

4. Conclusion

SISSvoc provides a lightweight search and retrieval API for RDF datasets based on SKOS. SISSVoc provides an abstracted view for end users and client applications to discover and access SKOS vocabulary resources much like any other web application without necessarily knowing any of the underlying technologies and semantic web languages used. The current design of SISSVoc (v3) is based on the Linked Data API and the examples described are implemented by configuring a Linked Data API endpoint. Since the triple-store hosting the content is coupled to the SISSVoc layer through a (usually public) SPARQL endpoint, this enables a flexible deployment pattern, in which multiple interfaces to the content are published, each one used as the basis of the next higher interface. This supports a range of application approaches. The Linked Data API provides significant capability out-of-the-box, including content negotiation for both human interfaces and machine readable interfaces. We have also presented a discussion outlining a number of SISSVoc deployments and evaluated it based on a discussion of its use in the environmental domains and some client applications, such as the WDTF Validation Service and the SISSVoc search tool.

5. Acknowledgements

Jackie Stewart (nee Githaiga) (CSIRO) for development work on earlier versions of SISSvoc; Stuart Williams & Chris Dollin (Epimorphics) for advice and debugging during the initial ELDA implementation.

6. References

[1] T. Baker, S. Bechhofer, A. Isaac, A. Miles, G. Schreiber, E. Summers, Key choices in the design of Simple Knowledge Organization System (SKOS), Web Semant. Sci. Serv. Agents World Wide Web. 20 (2013) 35–49.

[2] T. Berners-Lee, Linked Data - Design Issues, W3C Des. Issues. (2006).

[3] C. Bizer, R. Cyganiak, D2r server-publishing relational databases on the semantic web, in: 5th Int. Semant. Web Conf., 2006: p. 26.

[4] C. Bizer, T. Heath, T. Berners-Lee, Linked Data - The Story So Far, Int. J. Semant. Web Inf. Syst. 5 (2009) 1–22.

[5] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, et al., DBpedia - A crystallization point for the Web of Data, Web Semant. 7 (2009) 154–165.

[6] K.M. Cohen, S.C. Finney, P.L. Gibbard, J.-X. Fan, The ICS International Chronostratigraphic Chart, Episodes. 36 (2013) 199–204.

[7] S.J.D. Cox, K. Mills, F. Tan, Vocabulary services to support scientific data interoperability, in: Geophys. Res. Abstr. 15, Proc. EGU Gen. Assem. 2013, European Geoscience Union, 2013.

[8] S.J.D. Cox, S.M. Richard, A geologic timescale ontology and service, Earth Sci. Informatics. (2014).

[9] R. Cyganiak, C. Bizer, Pubby – A Linked Data Frontend for SPARQL Endpoints, (2009).

[10] P. Davidson, Designing URI Sets for the UK Public Sector, London, 2009.

[11] K. Degtyarenko, P. de Matos, M. Ennis, J. Hastings, M. Zbinden, A. McNaught, et al., ChEBI: a database and ontology for chemical entities of biological interest., Nucleic Acids Res. 36 (2008) D344–50.

[12] EIONET, GEMET Thesaurus, (2012).

[13] Epimorphics, Elda: the linked-data API in Java, (n.d.).

[14] R.T. Fielding, R.N. Taylor, Principled design of the modern Web architecture, ACM Trans. Internet Technol. 2 (2002) 115–150.

[15] J. Githaiga, G. Duclaux, S.J.D. Cox, J. Yu, Spatial Information Services Stack (SISS) Vocabulary Service – A Tool For Managing Earth & Environmental Sciences Controlled Vocabularies., in: eResearch Australas. 2010, 2010.

[16] S. Harris, A. Seaborne, SPARQL 1.1 Query Language, (2013).

[17] J. Hastings, P. de Matos, A. Dekker, M. Ennis, B. Harsha, N. Kale, et al., The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013., Nucleic Acids Res. 41 (2013) D456–63.

[18] A. Leadbetter, R. Lowry, D.O. Clements, The NERC Vocabulary Server: Version 2.0, in: Geophys. Res. Abstr. 14, Proc. EGU Gen. Assem. 2012v, European Geoscience Union, 2012.

[19] N.A.A. Manaf, S. Bechhofer, R. Stevens, The Current State of SKOS Vocabularies on the Web, in: Semant. Web Res. Appl., Springer, Berlin Heidelberg, 2012: pp. 270–284.

[20] A. Miles, S. Bechhofer, SKOS Simple Knowledge Organization System Reference, (2009).
[21] K.S. Nagendra, O. Bukhres, S. Sikkupparbathyam, M. Areal, Z.B. Miled, L.M. Olsen, et al., NASA Global Change Master Directory: an implementation of asynchronous management protocol in a heterogeneous distributed environment, in: Proc. 3rd Int. Symp. Distrib. Objects Appl., IEEE Comput. Soc, 2001: pp. 136–145.
[22] G. Ogg, Geologic TimeScale Foundation - Stratigraphic Lexicons, (n.d.).
[23] L.M. Olsen, G. Major, K. Shein, J. Scialdone, S. Ritz, T. Stevens, et al., NASA/Global Change Master Directory (GCMD) Earth Science Keywords. Version 8.0.0.0.0, (2013).
[24] E. Prud'hommeaux, A. Seaborne, SPARQL Query Language for RDF, (2008).
[25] S. Rajbhandari, J. Keizer, The AGROVOC Concept Scheme : A Walkthrough, J. Integr. Agric. 11 (2012) 694–699.
[26] L. Richardson, S. Ruby, RESTful Web Services, O'Reilly Media, 2008.
[27] F.B. Rogers, Medical Subject Headings (MeSH), Bull. Med. Libr. Assoc. 51 (1963) 114–116.
[28] L. Sauermann, R. Cyganiak, Cool URIs for the Semantic Web, (2008).
[29] J. Sheridan, J. Tennison, Linking UK Government Data, in: C. Bizer, T. Heath, T. Berners-Lee, M. Hausenblas (Eds.), Proc. Linked Data Web Work., CEUR, Raleigh, North Carolina, USA, April 27, 2010, 2010.
[30] P. Stickler, CBD - Concise Bounded Description, (2005).
[31] E. Summers, A. Isaac, C. Redding, D. Krech, LCSH, SKOS and Linked Data, Web Semant. Sci. Serv. Agents World Wide Web. 20 (2013) 35–49.
[32] J. Tuominen, M. Frosterus, K. Viljanen, E. Hyvönen, ONKI SKOS Server for Publishing and Utilizing SKOS Vocabularies and Ontologies as Services, Semant. Web Res. Appl. Lect. Notes Comput. Sci. Vol. 5554. (2009) 768–780.
[33] K. Viljanen, J. Tuominen, E. Mäkelä, E. Hyvönen, Normalized Access to Ontology Repositories, in: Proc. Sixth Int. Conf. Semant. Comput. (IEEE ICSC 2012), IEEE Press, Palermo, Italy, September 2012., 2012.
[34] R. Woodcock, B.A. Simons, G. Duclaux, S.J.D. Cox, Auscope's use of standards to deliver earth resource data, in: Geophys. Res. Abstr. 12, Proc. EGU Gen. Assem. 2010, European Geoscience Union, 2010.
[35] J. Yu, S.J.D. Cox, G. Walker, P.J. Box, P. Sheahan, Use of standard vocabulary services in validation of water resources data described in XML, Earth Sci. Informatics. 4 (2011) 125–137.
[36] ICD-10-CM - International Classification of Diseases, Tenth Revision, Clinical Modification, (n.d.).
[37] Medical Subject Headings (MeSH), (n.d.).
[38] Eurovoc, the EU's multilingual thesaurus, (n.d.).
[39] GEMET WebService API, (n.d.).
[40] NERC Vocabulary Server version 2.0 (NVS2.0) at BODC, (n.d.).
[41] Coastal and Marine Spatial Planning Vocabularies, (n.d.).
[42] linked-data-api - API and formats to simplify use of linked data by web-developers, (n.d.).