21st International Congress on Modelling and Simulation, Gold Coast, Australia, 29 Nov to 4 Dec 2015
www.mssanz.org.au/modsim2015

The Keyword Aggregator web service — a tool and methodology for managing digital objects' keywords

D. Benn [a], **N. J. Car** [b], **J. Yu** [b] **and S. J. D. Cox** [b]

[a] *CSIRO Information Management & Telecommunications, Adelaide, Australia*
[b] *CSIRO Land & Water Flagship, Brisbane & Melbourne, Australia*
Email: David.Benn@csiro.au

Abstract: The need to tag digital objects in various repositories such as datasets or scientific papers to allow for thematic classification is a well-known problem. In some cases a free tagging approach is used where contributors can add any keywords they desire while in other cases, selection from a controlled vocabulary is mandated.

To enable repository managers to allow both free tagging and the use of controlled vocabularies, we have created the Keyword Aggregator (KWA) system, which consists of the following components:

- A web service that:
  - provides fast search access to vocabulary content;
  - stores multiple, controlled, vocabularies of terms;
  - permits the addition of new keywords.
- An example widget that makes use of the web service and can be embedded in a web page.
- Use of a relational database that captures keyword use statistics.
- A management methodology that:
  - allows particular vocabularies to be selected for use on a per-widget instance basis;
  - allows vocabulary updating through storage in versioned repositories;
  - stores vocabularies with vocabulary-level metadata enabling vocabulary discovery;
  - permits one vocabulary to link to a single term in another vocabulary (allows a 'vocabulary-of-vocabularies').

Formal vocabulary representation systems such as the Simple Knowledge Organization System (SKOS) (Miles and Bechhofer, 2009) have been endorsed by the World Wide Web Consortium (W3C) where complex term relationships need to be represented. Keywords and whole-of-vocabulary metadata in KWA are stored as SKOS graphs in a graph database. Identification and, where necessary, creation or conversion of an initial set of science keyword vocabularies in SKOS has also been part of the work.

Although the focus of KWA is on the creation of a flexible keyword aggregation, management and search system for science keywords, vocabularies of any kind could, in principle, be handled by this system.

We describe our Keyword Aggregator system, which is aimed at lowering the effort required to provide digital object controlled vocabulary and folksonomy term tagging functionality to repository users. We present usage scenarios relating to CSIRO's Data Access Portal (DAP) and the MODSIM2015 paper system, key design goals, implementation details, current progress and future work.

Keywords: *Aggregation, keyword, vocabulary, RDF, SKOS, triple store*

1 INTRODUCTION

While it is well understood that there is utility in classifying digital objects using terms from controlled vocabularies, such vocabularies necessarily contain domain knowledge in the metadata provided for each term (e.g. definitions) and in the structure of the vocabularies themselves which describe how concepts are linked. This knowledge can take much time and effort to codify. When a term use scenario calls for terms from multiple domains, not only does the effort to assemble and use the corresponding multiple vocabularies rise but compounds as the knowledge required for vocabulary assembly in different domains may call for very different areas of expertise.

For these reasons, a system that aggregates already assembled vocabularies containing much domain knowledge and presents them for use is of *prima facie* utility for system implementers wishing to provide for the tagging of digital objects with terms. Such a system, by enabling vocabulary reuse, would not only save implementation effort but also promote term and whole vocabulary reuse. This brings benefits to all users since wider use of a term or vocabulary promotes better understanding.

Repository owners wishing to leverage the knowledge of their users in digital object classification may enable free tagging, sometimes known as 'folksonomy' (Peters and Becker, 2009) use. Such an approach has the advantage of not constraining a user to existing terms (which may be problematic if existing term sets are limited or inappropriate). However, this has disadvantages associated with aspects of the well-known 'vocabulary problem' (Furnas et al., 1987) such as poor differentiation between synonyms, plural/singular confusion and so on, stemming from the shortcomings of using undefined terms and human input. Strategies for dealing with folksonomy issues are also known: they can involve some interface tooling, such as term prompting implemented by the Joomla! Content Management System[1], nevertheless, costly forms of term curation are usually required[2].

Controlled vocabularies in SKOS and term-list, growing, folksonomies can be stored together within a single database that supports the Research Description Framework (RDF) (Prudhommeaux and Seaborne, 2008). While presented as a single database for searching, they can be maintained as separate entities using isolated namespaces. This allows for individual vocabulary/folksonomy management within the whole, including new vocabulary addition.

2 USAGE SCENARIOS

Examples of two digital object tagging scenarios we studied in order to form requirements for this project are CSIRO's Data Access Portal (DAP)[3] and the MODSIM2015 conference (to which this paper was submitted) paper submission system. Neither controlled vocabularies nor folksonomies have been used in these contexts for general keyword entry, although DAP does mandate the use of ANZSRC Fields of Research[4] codes for classification.

One could conclude from these scenarios that:

- such systems are not needed for digital object discovery, or
- the cost of system implementation is too high, or
- the priority of having a searchable keyword source is not high.

The KWA is for use by systems such as the DAP and MODSIM's paper submission system to replace their free text entry with a searchable source of keywords. It was understood that choice as to particular vocabularies for use — whether folksonomies and/or controlled vocabularies can be used

[1] https://docs.joomla.org/J3.x:How_To_Use_Content_Tags_in_Joomla!
[2] The popular technical advice site stackexchange.com explains their content tagging policy at https://blog.stackexchange.com/2010/08/tag-folksonomy-and-tag-synonyms/ and their 'tips for use' at http://meta.stackexchange.com/questions/18878/how-do-i-correctly-tag-my-questions demonstrates the effort required.
[3] http://data.csiro.au/dap
[4] http://www.abs.gov.au/ausstats/abs@.nsf/0/6BB427AB9696C225CA2574180004463E

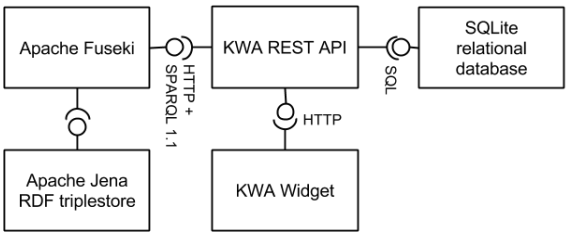

Figure 1. KWA component diagram with important interface types noted.

and whether or not to allow keyword addition by users — all needed to be configurable so that one tool may cater for multiple needs.

3 KEY DESIGN GOALS

The key design goals for KWA, informed by the usage scenarios initially considered, were:

- Fast keyword search that remains fast even with a large number of vocabularies;
- Relevant search results: semantic weighted and other search strategies needed to be catered for, not just full-text search. Possibly a choice of search strategies chosen by users;
- A web service exposing all tool functionality for ease of access;
- A web service demonstration client that could either be used directly or used as a reference implementation;
- Addition of user-defined keywords to allow for folksonomy-style use;
- Recording of keyword usage statistics both for the author's knowledge but also to potentially inform search strategies;
- Simple management of separate vocabularies to enable many to be used with little effort.

4 IMPLEMENTATION

4.1 Components

The KWA consists of five components as shown in Figure 1:

- **REST API**: A Web Service, written in Python Flask[5] and delivering JSON[6] content to users;
- **Example widget**[7] to act as a reference Web Service consumer that:
  - provides a reusable component for embedding in a web page, or
  - acts as a reference implementation for use with a language and framework of choice.
- **RDF triplestore**: For vocabulary storage. Apache's Jena[8] is currently used;
- **RESTful triplestore interface**: Apache's Fuseki[9] is used but any SPARQL1.1-compliant[10] API could be used;
- **Relational database**: For storing keyword usage information. Currently SQLite[11] is used.

4.2 Web Service

The KWA web service provides the following functionality:

[5] http://flask.pocoo.org
[6] http://json.org
[7] http://scikey.org/page/widget
[8] https://jena.apache.org/
[9] http://jena.apache.org/documentation/fuseki2/
[10] http://www.w3.org/TR/rdf-sparql-query
[11] https://www.sqlite.org/

- Search for keywords across aggregated vocabularies with vocabulary choice and search strategy given to the user;
- Vocabulary details from whole-of-vocabulary metadata;
- Vocabulary addition;
- Vocabulary metadata recording;
- Addition of new (user-defined) keywords;
- Storage and delivery of keyword usage information.

Detailed web service documentation is available online at http://scikey.org/page/documentation for each of the REST API functions.

Search. Search is this tool's *raison d'être* and its most complex function. The API provides an endpoint to which character strings can be submitted for searching with and which is configurable with respect to which vocabularies are included and the search strategy used.

Individual vocabularies within the KWA graph database are stored in individual *named graphs* so that vocabulary selection is enabled by simply selecting vocabularies and searching against their graphs only.

The search algorithm used is chosen by widget implementers by setting variables in widget config. Currently three algorithms can be selected for use. In all cases, the SPARQL query language (Prud'hommeaux and Seaborne, 2008) is used to query the graph database:

- simple term search - searches terms' labels only;
- weighted semantic search - searches amongst terms' SKOS properties and weights matches based on property relations;
- hierarchical search - finds matches traversing the broader and narrower relationships between SKOS terms.

Simple search is made available for comparison only and is not considered a viable search strategy as it ignores much of the knowledge contained within SKOS vocabularies. Weighted search is akin to a full-text search but one that preferences text matches in the primary descriptors of terms — such as term labels — over matches in secondary descriptors, such as descriptions and alternative labels. Terms with greater weights are returned before those with lower weights. A term matching a search in multiple fields has its multiple matches summed to provide a greater total weight. Search weighting can be tuned to alter the result order so it may be adjusted based on user feedback and testing.

Hierarchical search uses hierarchical relationships in SKOS to determine both the narrowest (most specific) match for a search and also broader matches that may not have been found via a text matching search. An example is that a search for 'sea' could return both the narrow 'Tasman Sea' and the broader 'water body'. The latter result may not contain the text 'sea' but is related to sea in the hierarchy.

This project is now comparing results between keyword search strategies and intends to further tune the algorithms used. It is expected that at least the hierarchical search may degrade with scale without significant engineering.

The values returned via search are the terms' label, and its URI. Other term values are not returned as they are resolvable via the term's URI. Metadata about the vocabulary from which a term codes is returned to enable users to make decisions about term validity based on vocabulary ownership or other facts about the vocabulary imparted via metadata.

To meet the design goal of providing fast search (results returned in 'web time', i.e. the time a web user will realistically accept which is less than a second) for all search strategies and as content grows, we have considered the following methodology: a. tuning of the search algorithms (SPARQL

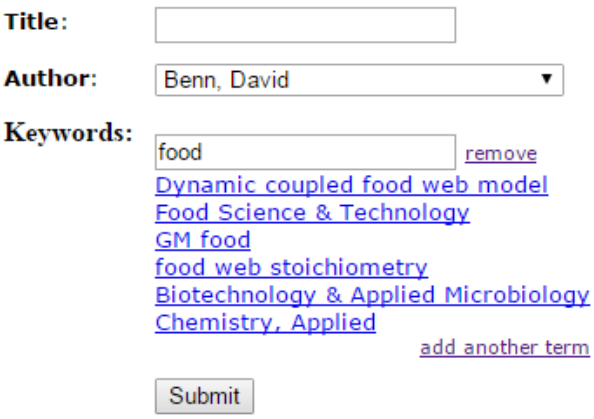

Figure 2. KWA widget in use with the term 'food' searched for. Due to the use of a weighted search strategy search non-label elements of terms, such as term descriptions, some of the found terms do not contain 'food' in their title.

queries), b. pre-computing relationships between terms (required for the hierarchical search strategy), c. dedicating more resources to the graph databases implementing the searches, d. moving to a more powerful graph database (the StarDog[12] product claims greater performance than our current Jena implementation), e. implementing a map-reduce-style search platform, perhaps based on the Apache Spark project using the GraphX package[13] project. The current KWA system allows for all of these changes without any disruption to the user experience due to the API layer's abstraction from the database layer.

A complete description of the search algorithms is available at http://scikey.org/page/documentation.

4.3 Widget

The KWA widget is a consumer of the KWA API. Widgets present as a partial HTML form that can be embedded within a host system form for use. Figure 2 shows the KWA example widget in use. The example widget is written in HTML and JavaScript with substantial use of the jQuery[14] JavaScript toolkit. Communication with the RESTful web service is via asynchronous HTTP requests with JSON payloads passed to and from the web service.

The widget allows search and selection of keywords relating to a digital object. When 3 or more characters have been entered into the *Keywords* text box, a REST API keyword search is initiated. The JSON result is processed to yield a list of links. If the user selects a link from this list, the list is collapsed and the keyword added to a list for submission. The user may go on to search for more keywords and select from the search results before submission. Individual selected keywords can also be removed. Selected keywords are presented to the system hosting the widget as simple HTML form textbox values of URIs, one for each selected keyword. The host system need not store information other than the term's URI as all KWA terms have resolvable (dereferencable) URIs. If no keywords are found from a search string, the user is given the option to add a new keyword along with an optional description, as shown in Figure 3. Navigating away from the page (usual for host system form submission) causes keyword usage information to be stored in the relational database via a REST API call.

The widget is configurable so that a widget host may select only certain vocabularies in the KWA database for use. Additionally, certain search strategies may be selected. Where the widget is not able to be used directly by a host system (perhaps due to code environment incompatabilities) the

[12] http://stardog.com
[13] http://spark.apache.org/graphx/
[14] http://jquery.org

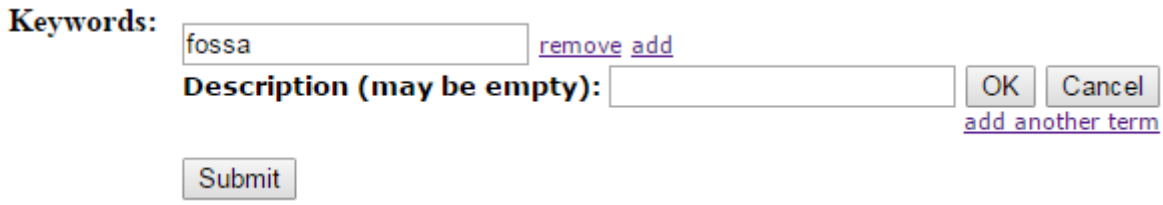

Figure 3. User defined keyword addition. The term 'fossa' is being searched for and no matches are found. It, or an extension of it entered by the user, may be stored as a user-defined keyword, as per folksonomy practice. A description for the term may be entered.

example widget can be used as a reference implementation. Complete details of how to use, configure and deploy the example widget are available online[15].

4.4 Vocabularies

Within this project we collated a compendium of vocabularies[16] potentially useful for keyword searching. Vocabularies for use by the KWA must be formalised in SKOS. This does limit vocabulary selection to a subset of those in our compendium but without a single data model such as SKOS specified, multiple vocabulary searching becomes infeasibly complex. Many important vocabularies have already been formalised in SKOS, such as the Library of Congress' Subject Headings[17] and for some we deemed important not already in SKOS, such as Thompson Reuters Science Citation Index[18], we undertook conversion to SKOS.

Historical keyword data from previous Modelling and Simulation Society of Australasia's MODSIM conferences (2011 & 2013) was also processed into SKOS for potential use by a widget deployed for future MODSIM conferences. Unlike the Thompson-Reuters terms, this was not already presented as a vocabulary but first had to be mined from published papers. We are working on a process to allow vocabularies to be submitted to the system by users.

In order to manage vocabularies, we developed a vocabulary metadata schema[19] based largely on the VOAF schema[20] but with extensions to allow for vocabulary linking. Vocabulary linking, is one of the novel aspects of this project. Our metadata schema for vocabularies allows the root term of one vocabulary to be associated with a single term in another vocabulary thus generating a 'vocabulary-of-vocabularies' or a hierarchy of vocabularies which we have described in detail previously (Car et al., 2015). This does not prevent, and we encourage, more fine-grained inter-vocabulary term linking as per normal SKOS vocabulary formulation.

Keywords submitted by users are placed in a 'user-defined' vocabulary that stores terms in SKOS but without any relationships between terms. In some instances, future curation could see terms moved into more formal vocabularies however URIs for terms, once created, will not change allowing for seamless term use even when moved.

See http://scikey.org/page/vocabularies for information about loaded vocabularies and https://stash.csiro.au/projects/VOC/repos/kwag-vocabs/ for the SKOS vocabularies used (in the RDF Turtle format).

4.5 Usage Statistics

In any system that is used, it is good practice (essential even) to capture usage statitics. In the KWA's

[15] http://scikey.org/page/widget
[16] http://scikey.org/page/documentation
[17] http://id.loc.gov/authorities/subjects.html
[18] http://ip-science.thomsonreuters.com/mjl/scope/scope_scie
[19] http://scikey.org/def/vocab
[20] http://lov.okfn.org/vocommons/voaf/v2.3/

case, we may be able to determine relative keyword utility where there are two similar keywords and such knowledge may be used in a future search strategy. Since usage statistics could also indicate keyword proximity if a user selects multiple similar terms, future clustering learning may take place. We are not able to use the primary graph database to store these statistics since we wish to leave the vocabularies stored there in the form in which we receive them as we are not their creators. It would also be inefficient to use a graph database to store tabular data such as usage statistics.

5 FUTURE WORK

A number of improvements and enhancements are possible and await resourcing:

- Good search performance with large vocabularies and a large number of vocabularies;
- A search strategy using inter-vocabulary individual term linking (e.g. *skos:related*);
- Provide term metadata (e.g. description) via the widget to enhance keyword selection;
- The automated ingestion of a large number of known vocabularies;
- Establish a streamlined process for vocabulary submission;
- Enhanced vocabulary management through federated governance.

6 CONCLUSIONS

System implementers of the Keyword Aggregator (KWA) are able to provide a web or desktop form interface for keyword selection to users of their system. They can do this without needing to:

- find, store and manage a large array of published vocabularies;
- maintain relations between vocabularies of broader and narrower concepts;
- managing the collection of new informal terms not yet placed into a controlled vocabulary;
- ensure term search performance while the stored term base grows.

The KWA reduces the effort required by digital object management system implementers in providing access to the large body of knowledge already captured in existing formal vocabularies to manage vocabularies approved for use in a particular scenario and domain.

ACKNOWLEDGEMENT

The authors acknowledge the CSIRO Scientific Computing eResearch collaboration proposal that led to a software development resource being made available for this work.

REFERENCES

Nicholas John Car, Simon Cox, Jane Frazier, Benn David, and Jonathan Yu. A ”vocabulary-of-vocabularies” with tools for users. 2015. URL https://eresearchau.files.wordpress.com/2015/07/eresau2015_submission_16.pdf.

G. W. Furnas, T. K. Landauer, L. M. Gomez, and S. T. Dumais. The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971, 1987. ISSN 00010782. doi: 10.1145/32206.32212.

Alistair Miles and Sean Bechhofer. SKOS Simple Knowledge Organization System Reference, 2009. URL http://www.w3.org/TR/2009/REC-skos-reference-20090818/.

I. Peters and P. Becker. *Folksonomies: Indexing and Retrieval in Web 2.0*. Knowledge & information : studies in information science. De Gruyter/Saur, 2009. ISBN 9783598251795. URL http://www.degruyter.com/viewbooktoc/product/42362.

Eric Prudhommeaux and Andy Seaborne. SPARQL Query Language for RDF, 2008. URL http://www.w3.org/TR/rdf-sparql-query/.