Publishing and Exploiting Vocabularies using the OpenSKOS Repository Service

Hennie Brugman [1], Mark Lindeman [2]
1. Meertens Institute
P.O. Box 94264, 1090 GG Amsterdam
E-mail: hennie.brugman@meertens.knaw.nl

2. Pictura Database Publishing
De Hoefsmid 11, 1851 PZ Heiloo
E-mail: M.Lindeman@pictura-dp.nl

Abstract

Many vocabularies in eHumanities and eCulture domains can, and increasingly often are converted to SKOS. The OpenSKOS web service platform provides easy ways to publish, upload, update, harvest, query and distribute SKOS vocabulary data. This has benefits for vocabulary builders, vocabulary consumers and builders of tools that exploit vocabularies. In this paper we present and discuss the OpenSKOS system and a number of its applications, including an application from the domain of linguistic resources and tools.

1 Introduction

The application and relevance of vocabularies for the description of cultural heritage and scientific collections is making a comeback. One of the motivators for this comeback is the emergence of Semantic Web and Linked Open Data. There is much interest in application of data and text mining techniques to disclose collections, but it turns out that many of these techniques also build on vocabulary information.
Recent years have seen forms of standardization for vocabulary data that are consistent with Semantic Web and Linked Data principles. Well known is the W3C SKOS (Simple Knowledge Organization System) recommendation (Miles, 2009). More and more vocabularies, especially in the cultural heritage domain are mapped and converted to the RDF-based SKOS format and data model.

In 2004 the Dutch CATCH research programme started. CATCH (Continuous Access To Cultural Heritage) consists of a number of projects that do research regarding computer science and humanities research questions that are driven by cases from daily practice at large Dutch cultural heritage institutions. CATCHPlus is a partner project of CATCH that does valorization: it has the assignment to turn research prototype systems and demonstrators from the CATCH programme into tools and software services that can actually be used by cultural heritage professionals and users.

CATCHPlus tools and services should, where possible, contribute to the emerging infrastructure for digital cultural heritage. One aspect that many of the tools and services in CATCHPlus have in common is that they deal with or exploit vocabulary data. Therefore CATCHPlus stimulated standardisation of vocabulary formats to SKOS and also started work on a shared service that adds some standardisation to the way these SKOS vocabularies are made available and accessed: OpenSKOS[1], a web service based vocabulary publication platform.

Section 2 will describe requirements and motivations for OpenSKOS. Section 3 will describe the OpenSKOS architecture and components in detail, section 4 will position OpenSKOS in comparison with the ISOcat terminology service and with Linked Open Data. Section 5 describes current and future applications and clients of the OpenSKOS service. We will end the paper with an evaluation and conclusions (section 6).

2 Problem statement

The importance of and interest in vocabulary resources is increasing. These resources are typically created in specialized vocabulary maintenance tools or in modules of collection management systems. They are made available online using interactive web applications or in the form of Linked Data at the most. Over the last couple of years some standardization with respect to format has taken place: many vocabularies are currently mapped to SKOS.
However, it is often still a cumbersome process to locate suitable vocabularies and to (re)use them for one's own resource description tasks, in one's own tool environment. This is especially true when a vocabulary is well maintained and therefore frequently updated. To use a concept that is newly introduced by the vocabulary editors typically requires export and upload/download of the full vocabulary, proprietary format conversions and software adaptation or configuration steps by the producers of several collection management systems.

[1] http://openskos.org

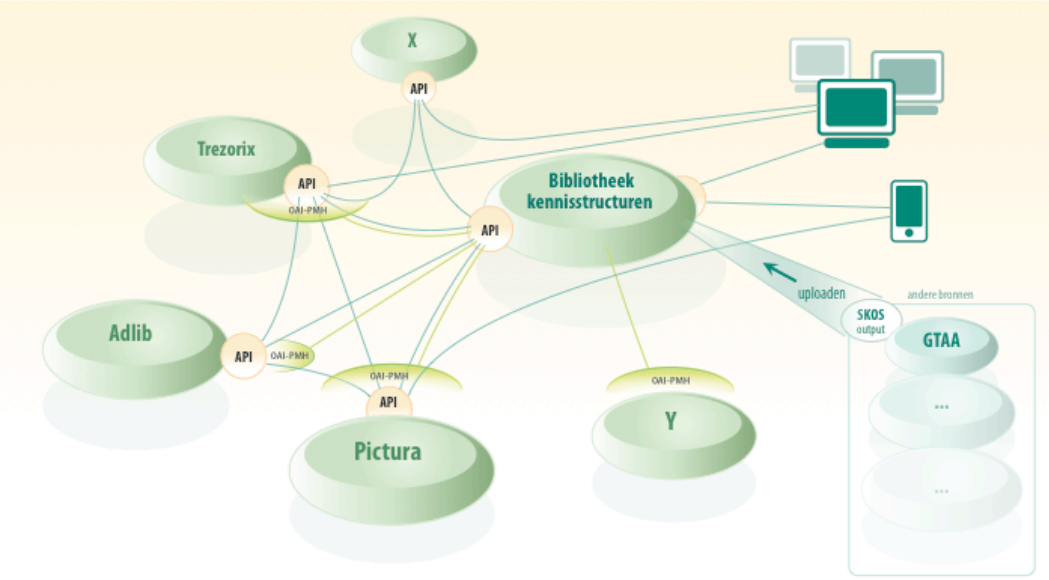

Figure 1: OpenSKOS architecture

Some web service based solutions also provide access to vocabularies as data, but these often have other shortcomings. They do not support periodic and/or incremental updates, they do not support the full underlying data model of the vocabularies (e.g. they are not able to handle relations between concepts), or they are optimized for other use cases than providing concepts for resource description (e.g. they have no proper support for handling long lists of entity names).

The Linked Data movement also imposes additional requirements on vocabulary services: concepts should be identified with stable, resolvable http URIs. Content negotiation is a desirable feature for a Vocabulary service.

Finally, web based (Open) Annotation (Sanderson, 2011) is a new development, that also imposes linked data type of requirements on Vocabulary services. It should be possible to annotate a web resource with URIs of concepts in online repositories.

3 The OpenSKOS service

OpenSKOS is a web service based approach to publication, management and use of vocabulary data that can be mapped to SKOS. The name is not meant to suggest that SKOS is not open; it refers to 'infrastructure and services to provide *open access* to SKOS data'. The main objective is to make it easy for vocabulary producers to publish their vocabularies and updates of it in such a way, that they become available to vocabulary users automatically and instantaneously, and independent of the specific software tools of these vocabulary users.

3.1 Architecture

Figure 1 shows the OpenSKOS architecture, which is a peer-to-peer architecture. Several sites can run instances of the freely available OpenSKOS repository software. Peers with a more centralized role are not technically necessary, although not excluded. Each site can be accessed by means of a RESTful API (Richardson, 2007) that supports a range of queries to retrieve or update SKOS vocabulary information in the repository. Having local copies of vocabularies in a repository instance implies that these can be searched efficiently on basis of locally created indexes.

Different OpenSKOS sites can exchange local copies of vocabularies using the OAI-PMH[2] protocol: OpenSKOS has built-in OAI-PMH data providers and harvesters. New vocabularies can be imported into the system in several ways: they can be harvested from another instance of OpenSKOS, they can be harvested from external OAI data providers, they can be included by implementation of the OpenSKOS API by other parties, or they can be uploaded using a built-in upload module.

Finally, OpenSKOS software contains a Dashboard to support a number of management tasks on each instance of OpenSKOS. This Dashboard can only be accessed after successful authentication.

3.2 The OpenSKOS RESTful API

The system's API is defined in a collaborative effort between the CATCHPlus project office, three major commercial tool providers for the Dutch Cultural Heritage sector (Adlib Systems, Pictura Database Publishing and Trezorix) and the Rijksdienst voor het Cultureel Erfgoed (Dutch department for cultural heritage). The specification is based on previous experiences and known use cases of all partners. The W3C SKOS recommendation was taken as the underlying data model.

2.3.1 Functional scope of the API

To start with, the API can resolve (skos) Concepts and ConceptSchemes ('vocabularies') by URI in a number

[2]http://www.openarchives.org/OAI/openarchivesprotocol.html

of representation formats (JSON, RDF/XML, html). This implies that Linked Data access is a sub set of the web services functional scope. The resolve API has query parameters that allow filtering on language used, and specification of what information is/is not included in the result.
Second, the API has 'find' functionality for Concepts and ConceptSchemes. It supports a query parameter 'q' that takes queries according to the Apache Lucene Query Parser Syntax as values. Searching is possible over all SKOS based fields and over Dublin Core (dcterms) fields, if those are present. The result of a 'find' query is a list of Concepts (represented in the same way as for the concept resolve) and a diagnostics block, for example with number of results that match and number of results on page. Paging and sorting of results is supported.
A specialization of the /find API is the OpenSKOS 'auto complete' function, meant for interactive searching for matching concept labels starting with some characters. The primary use case for this auto complete is supporting resource description tasks in some collection or metadata management system.
The OpenSKOS API namespace contains *Collections* and *Institutions* that are not part of the SKOS model but added for practical reasons. Collections can group a number of conceptschemes together that constitute one resource from an organisational/data management perspective. For example, the thesaurus of the Netherlands Institute for Sound and Vision (archive of the Dutch public broadcast corporations) consists of six sub thesauri but is maintained and published as a whole. *Institutions* are added to make information available on the vocabulary publishers themselves, and to associate authorized vocabulary managers with.
The API explicitly covers SKOS properties that are used to define mappings between concepts, also mappings between concepts belonging to different conceptschemes. The OpenSKOS repository is also a place where mappings across vocabularies can be maintained and exploited.
The OpenSKOS API not only supports HTTP GET operations on the resources described before, but for many of those resources it also supports PUT, POST and DELETE operations. It is therefore possible to perform vocabulary maintenance tasks directly on the repository using the API. For REST examples see openskos.org.

The CATCHPlus project office and Pictura together have built an OpenSKOS implementation that includes an implementation of the API. This implementation is internally based on Apache SOLR. It also includes implementations of other OpenSKOS components: a Dashboard, OAI harvester and data provider (including a job scheduler) and upload module for SKOS uploads.

3.3 OAI-PMH and upload modules

There are in principle three ways to enter vocabulary data into the OpenSKOS repository: create it from scratch using the APIs PUT and POST operations, upload it using the built-in upload module or harvest it using the built-in OAI-PMH harvester and job scheduler.
OpenSKOS repositories are able to harvest vocabulary data or to provide harvesting access to specific vocabularies from other OpenSKOS instances. This harvesting can be done periodically and incrementally. OpenSKOS includes a job scheduler that can be configured to run periodic harvesting jobs.
Reasons to harvest vocabularies to one's own OpenSKOS instance are: it can be used for an initial full download, and it subsequently keeps vocabulary information up to date. Another reason could be to maintain a copy for local indexing and searching. A reason to provide access for harvesting by others: most efficient, flexible and controlled way to allow downloads of potentially large data sets (http could lead to long download times and time outs).

OpenSKOS has a built-in upload module that can only be operated by authorized users using the system's Dashboard.

3.4 Dashboard

For management tasks by authorized users the system has an interactive Dashboard component. After successful authentication a user can access several panes. The "Manage institution" pane allows the user to enter and modify institution metadata, like name, contact information and website. "Manage collections" presents the user with an overview of available collections, and allows the user to create new ones. These collections are associated with the users' Institution. Each collection has associated metadata, like title, description, links to websites, and license information (preferably Open Database licences, of course). Also, for each collection it is possible to specify whether it is harvestable by other OpenSKOS instances and if the associated data is imported by upload or by OAI-PMH harvesting. In the latter case the OAI data providers' base URL can be specified.
Collections are the unit of 'upload' or 'maintenance', and can consist of data for several SKOS ConceptSchemes.
The "Manage users" pane gives an overview of existing users, their email addresses, their access rights (do they have writing access using the API, using the Dashboard or both) and their API key. It also supports creation of new users.
Finally, the "Manage jobs" pane gives an overview of scheduled and finished harvest and upload jobs.

Institution and collection info can not only be inspected and modified using the Dashboard; it is also available to anyone for inspection using the relevant API calls, represented as RDF/XML, JSON or html. The html representation makes it possible to browse over the repository content starting at an Institution, via its Collections and ConceptSchemes to representations of the Concepts themselves.

Figure 2: Snippets of user interfaces of OpenSKOS clients

3.5 Authentication and authorization

Since the main objective of OpenSKOS is to be 'open' we chose not to support authenticated 'read' access to the repository's content, all SKOS information is world-readable. In fact, we actively promote the use of open license forms like the Open Database license by offering this as an optional license form to creators of new vocabulary Collections.

For modification operations (create, update, delete) we support two levels of authorization: access using an API key, and access via the system's Dashboard. At API level modifications to Concepts and ConceptSchemes can be made. Modifications to Institutions, Collections and users all require authentication via the Dashboard.

Users can have either or both of the authorization levels.

4 Related work

OpenSKOS can in terms of genericity be positioned somewhere between a domain- and community-specific terminology repository solution as ISOcat and the generic and general purpose Linked Open Data approach.

ISOcat (Windhouwer, 2010) is an ISO TC 37 registry for Data Categories. These Data Categories are mainly intended for linguistic concepts. ISOcat by definition does not support relations between concepts and relies on separate relation registries for this. Main use cases for ISOcat are registration of concepts and providing a platform for standardisation of linguistic terminology. ISOcat therefore is not the optimal place to maintain or serve large lists of term labels. SKOS and OpenSKOS are less restrictive: they are not restricted to a certain domain, support relations between concepts and support a wider range of use cases. Representing and serving long term lists is normal practice. ISOcat has a RESTful web service that can be and actually is used to feed the OpenSKOS service (see chapter 5.3 about CLAVAS).

Linked Data on the other hand is even more generic: it is not restricted to vocabulary type of data, as SKOS and OpenSKOS are. It can represent any mix of data, metadata and concepts and links between those. The drawback is, that considered as a protocol it is much simpler than the ISOcat and OpenSKOS RESTful APIs. Linked Data access by means of resolvable and stable http URIs and support for content negotiation is a subset of the functionality of the OpenSKOS API.

5 Applications

The OpenSKOS repository service and architecture is the outcome of a process of several years, during which prototypes and experimental tools were built and tested. Over these years several academic, commercial and cultural heritage partners got involved. This section describes a bit of OpenSKOS' history and context, before it discusses current and planned applications of the system.

5.1 OpenSKOS history and context

Previous work in the CATCH research programme and in CATCHPlus resulted in a demonstrator and in a first version of the Vocabulary Repository service. This first version was implemented as a 'thin' Java layer on top of an RDF store (Openlink Virtuoso). Although stable and performant (e.g. online auto completion over the web works fine), this implementation makes a large demand on memory, and we had doubts about its scalability. Furthermore, its API is at best "REST-like", it has limited and incomplete support for modification operations, and there are no provisions for web upload, OAI-PMH harvesting or user authentication.

Nevertheless, this system was and is actually used for

daily collection description work by the triangle Netherlands Institute for Sound and Vision, National Archive, and Pictura and was found an elegant and interesting solution. (S&V is the thesaurus provider, National Archive does collection description with S&V terms using Pictura's Memorix tool).
This relative success led to intensive discussions between CATCHPlus, RCE, Adlib, Pictura, Trezorix that led to refinement of the OpenSKOS concept and a proper RESTful API specification that built on the knowledge, use cases and experience of all partners. Subsequently, the API, infrastructure and Dashboard were implemented by Pictura and CATCHPlus.
Due to this long history with frequent discussions, presentations and experiments in the Dutch cultural heritage context, there is now serious interest to participate. Several large Dutch CH institutions are currently involved in some way.
Recently CLARIN-NL also started a project to apply OpenSKOS for linguistic vocabulary data (see 5.3).

5.2 OpenSKOS clients

Some API clients already exist. A generic browse and search web application was built for CATCHPlus (by Q42, see figure 2). All access to vocabulary data used and shown in this web application is exclusively retrieved via API calls.
Pictura's collection management application Memorix is used on daily basis by National Archive for description of their online image collection. Memorix also functions as an OpenSKOS client.
Sound and Vision has started development of a web based thesaurus management application on top of the OpenSKOS editing APIs to manage their GTAA thesaurus.

5.3 Application by CLARIN(-NL): CLAVAS

Within the Dutch CLARIN context there turned out to be a need for an additional effort to promote uniform terminology. While ISOcat focuses on standardisation of sets of concepts (Datcats) there is an additional need for support of relative simple, but long lists of terms, especially in the context of metadata creation and editing. Therefore CLARIN-NL started the CLAVAS project, which is an application of OpenSKOS. The CLARIN project makes several contributions to OpenSKOS, and CLARIN in turn can benefit from additional efforts done for OpenSKOS. These contributions are three additional SKOS-ified resources (ISO 639-3 language codes, access to public parts of ISOcat through the OpenSKOS API and architecture, and a vocabulary of organisation names relevant for the international domain of linguistic tools and resources. It is explored if this list can be bootstrapped by existing metadata descriptions containing organisation information.
An additional CLAVAS component is a simple web application that supports basic vocabulary curation tasks on simple concept lists.
The CLAVAS project is done by the Meertens Institute, which also hosts the central CATCHPlus project office.

6 Evaluation and conclusions

The OpenSKOS service can be consulted in many use cases where vocabularies play a role. Some examples :

- When defining a metadata component, as for example in the CMDI framework it is possible to associate a metadata field with a ConceptScheme in OpenSKOS simply by associating the field with the URI of the ConceptScheme.
- When creating metadata in a metadata editor values for fields can be selected using the auto complete API of OpenSKOS.
- The service can be exploited in several browse in search scenarios, for example for faceted browsing or for query formulation.
- When Concepts have labels in multiple languages, localized views of metadata records can be displayed.

OpenSKOS supports all SKOS relations between Concepts, both within vocabularies and across vocabularies. SKOS and OpenSKOS also support enrichment of vocabulary concepts with links to other resources on the web (more specifically, in the Linked Data cloud).

Probably the greatest benefit of OpenSKOS is that it provides an easy publication platform for all resources that can be 'SKOS-ified'. This has advantages for vocabulary publishers, for vocabulary consumers and for builders of tools that create or exploit vocabularies.

Advantages for vocabulary publishers are:
- Offering vocabularies to others is as easy as a simple upload action.
- It is easy to use your own vocabulary in the tools of others, if these tools use OpenSKOS.
- Vocabularies can easily and frequently be updated without involvement of others.
- It is easy to link your own vocabulary to vocabularies of others.

Advantages for vocabulary consumers :
- Easy discovery, evaluation and reuse of existing vocabularies (and therefore a reduced need to construct your own).
- New browse and search possibilities.
- Always up to date versions of vocabularies are available

Advantages for tool builders :
- No more periodic updates, no more specific adaptations for specific vocabularies.
- Can benefit from efforts of other tool builders and of vocabulary publishers.

- Can use OpenSKOS API functionality for a range of use cases.

OpenSKOS is available as open source from GitHub, and as installable package. It is implemented on basis Apache SOLR technology in a scalable way. A community of OpenSKOS users is already emerging.

7 Acknowledgements

We would like to thank all people and institutions that contributed to the realization of OpenSKOS by investing time, energy and/or funding. We especially would like to mention RCE, Adlib Systems and Trezorix for their contributions to the definition of the OpenSKOS architecture and API, and the funders of CATCHPlus: the Netherlands Organisation for Scientific Research (NWO), and the Dutch ministeries for Education (OCW) and Economic Affairs.

8 References

Miles, A., Bechhofer, S. (2009). SKOS Simple Knowledge Organisation System Reference. *W3C Recommendation* 18 August 2009.

Richardson, L., Ruby, S. (2007). *RESTful Web Services: Web services for the real world.* O'Reilly Media. May 2007.

Sanderson, R., Van De Sompel, H. (2011). Open Annotation. Beta Data Model Guide. http://www.openannotation.org/spec/. 10 August 2011.

Windhouwer, M.A., Wright, S.E., Kemps-Snijders, M. Referencing ISOcat data categories. *In proceedings of the LRT standards workshop* (LREC 2010), Malta, May 18, 2010