

MusicMash2: Mashing Linked Music Data via An OWL DL Web Ontology

Jeff Z. Pan, Stuart Taylor and Edward Thomas
Department of Computing Science, University of Aberdeen, UK

Abstract

In this paper we present MusicMash2, an ontology-based semantic mashup application. The two main problems involved in building semantic mashup applications of this type are, firstly, a lack of populated domain ontologies available on the Web and, secondly, the poor precision of standard search facilities provided by many folksonomy-based tagging systems. We show how our reusable infrastructure can address these two problems. Furthermore, to evaluate the benefits of our approach, we compare MusicMash2 to its predecessor MusicMash, which is based on standard Web 2.0 techniques.

Introduction

MusicMash2 is an ontology-based semantic mashup application, which is intended to integrate music related content from various folksonomy based tagging systems, linked open data¹, and music metadata Web services. MusicMash2 provides the functionality for users to search for tagged images and videos that are related to artists, albums, and songs.

An application of this nature presents two main problems. The first problem lies with the availability of populated domain ontologies on the Web. The Music Ontology (<http://www.musicontology.com/>) provides classes and properties for describing music on the Web; however, to instantiate the ontology, MusicMash2 must integrate music meta-data from various sources. The resulting populated ontology may be very large, so a suitable mechanism for reasoning over this ontology must be used – this mechanism must be scalable and efficient enough to handle queries on this ontology with an acceptable response time for users.

The second problem is that searching both within and across folksonomy based systems is an open problem. A naive approach to folksonomy search, such as those provided most tagging systems², results in unacceptable precision in domain specific searches. The lack of search precision is due to the limitations of tagging systems themselves (Passant 2007). MusicMash2 addresses this problem by making use of the Folksonomy

Search Expansion methods provided by the Taggr system (Pan, Taylor, & Thomas 2009), which makes use of the populated Music Ontology stored provided by MusicMash2. An alpha version of MusicMash2 is available at <http://www.musicmash.org/>.

In what follows, we first present our ontology infrastructure TrOWL, accompanied by ONTOSEARCH2 the ontological search engine and Taggr the ontology-based folksonomy search optimiser, and then show how to use such infrastructure to build semantic mashup applications like MusicMash2, which will then be compared to its predecessor MusicMash to illustrate the benefits that our infrastructure provides.

Reusable Infrastructure

Ontology Storing and Keyword Association

ONTOSEARCH2³ (Pan, Thomas, & Sleeman 2006; Thomas, Pan, & Sleeman 2007) is an ontological search engine that is based on the TrOWL⁴ ontology infrastructure, the basic components of which include an ontology repository, where users can submit and share ontologies; a query rewriter which rewrites user queries submitted in SPARQL so they can be executed on the repository; and some scalable ontology query engines. A useful feature of the TrOWL repository is that it automatically associates keywords (in values of annotation properties as well as implicit metadata in target ontologies) with concepts, properties and individuals in the ontologies. Default annotation properties include the `rdfs:label`, `rdfs:comment`, `rdfs:seeAlso`, `rdfs:isDefinedBy`, and `owl:versionInfo` properties; we also define the `dc:title`, `dc:description`, and `foaf:name` properties (from Dublin Core⁵ and FOAF⁶) as annotation properties. Implicit metadata is drawn from the namespace and ID of each artifact in the ontology.

These keywords are weighted based on ranking factors similar to those used by major search engines⁷.

³<http://www.ontosearch.org/>

⁴<http://trowl.eu/>

⁵<http://dublincore.org/>

⁶<http://www.foaf-project.org/>

⁷<http://www.seomoz.org/article/search-ranking-factors>

¹<http://linkeddata.org/>

²YouTube API: <http://www.youtube.com/dev/>

TrOWL uses these scores to calculate the *tf·idf* (Salton & McGill 1983) for each keyword found within the ontology, and normalises them using a sigmoid function such as the one shown in (1) to a degree between 0 and 1.

$$w(n) = \frac{2}{1.2^{-n} + 1} - 1 \quad (1)$$

Ontology Query Answering

Other basic components of TrOWL include its scalable OWL 2 QL and OWL DL query engines. Within TrOWL, ontologies to be loaded into its repository are reduced to OWL 2 QL⁸ (Calvanese *et al.* 2005) representations using a process called Semantic Approximation (Pan & Thomas 2007). This process allows conjunctive querying against the ontology (after some careful query rewriting) to be performed within a database, while still giving soundness guaranteed results for all queries, and sound and complete (w.r.t. the original ontology) results for database-style queries. Preliminary evaluation (Pan & Thomas 2007) shows that this query engine can scale to (at least) millions of individuals.

Ontology Uncertainty Handling

Being able to handle fuzzy and imprecise information is crucial to the Web. TrOWL also consists of a query engine for fuzzy OWL 2 QL (Pan *et al.* 2008). The query engine supports threshold queries and general fuzzy queries over fuzzy OWL 2 QL ontologies. Users of the query engine can submit fuzzy OWL 2 QL ontologies via the Web interface of ONTOSEARCH2⁶, and then submit f-SPARQL (Pan *et al.* 2008) queries, such as the following one, against their target ontologies.

```
#TQ#
PREFIX music: <http://musicmash.org/NS/>
```

```
SELECT ?x WHERE {
  ?x a music:MusicArtist .
  ?x a music:Popular . #TH# 0.7
  ?x a music:Active . #TH# 0.8
}
```

where #TQ# declares a threshold query, while #TH# is used to specify thresholds for atoms in the query. Therefore this query searches for an instance of `MusicArtist` which is a member of the class `Popular` with degree 0.7, and a member of the class `Active` with degree 0.8.

Preliminary evaluations shows that performance of the fuzzy OWL 2 QL query engine is in most cases close to the performance of the crisp OWL 2 QL query engine (Pan *et al.* 2008).

Ontology Searching

ONTOSEARCH2 is an ontological search engine that allows users to search its repository with keyword-plus-entailment searches, such as *searching for ontologies in*

⁸OWL 2 QL is also known as DL-Lite.

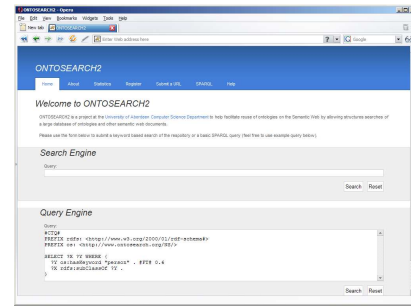


Figure 1: ONTOSEARCH2 Screenshot

which class *X* is a sub-class of class *Y*, and class *X* is associated with the keywords “Jazz” and “Rock”, while class *Y* is associated with the keyword “Album”. The search could be represented as the following threshold query:

```
#TQ#
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX os: <http://www.ontosearch.org/NS/>
PREFIX kw: <http://www.ontosearch.org/KW/>

SELECT ?x WHERE {
  ?x os:hasKeyword kw:jazz . #TH# 0.5
  ?x os:hasKeyword kw:rock . #TH# 0.7
  ?x rdfs:subClassOf ?y .
  ?y os:hasKeyword kw:album . #TH# 0.8
}
```

where `kw:jazz`, `kw:rock`, and `kw:album` are representative individuals for keywords “jazz”, “rock”, and “album”, respectively. The thresholds 0.5, 0.7, and 0.8 can be specified by users. The keyword-plus-entailment searches are enabled by the fuzzy DL-Lite query engine as well as the semantic approximation components.

The keyword-plus-entailment searches provided in ONTOSEARCH2 allows both TBox and ABox queries.

Relating Folksonomies to Ontologies

The *Taggr* system provides a simplified interface to ONTOSEARCH2 to perform useful operations that are related to folksonomy based systems. It stores a basic ontology (which we refer to as the “tagging database”) in the TrOWL repository, capturing the relationships between users, tags and resources in the folksonomy based systems it supports⁹.

Taggr allows users to provide a set of arbitrary resources and related tags to be added to the tagging database in the TrOWL repository. A Web service and traditional Web interface are provided so that users can interact with the tagging database without having to understand the internal representation used by the system.

⁹Taggr currently supports the YouTube and Flickr tagging systems.

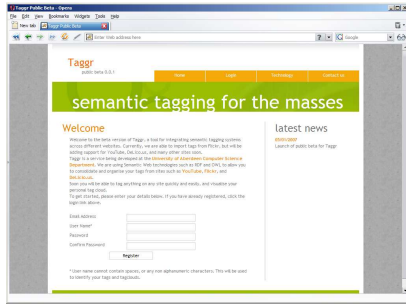


Figure 2: Taggr Beta Screenshot



Figure 3: MusicMash 1.0 Screenshot

Common Folksonomy Interface

Taggr also provides an ontology-enabled common interface for folksonomy based systems. It provides the functionality to gather resources and their related tags from the systems that it supports, and populate them to its tagging database from time to time. In case an application requests some resources that Taggr does not know about, it can simply make a call to Taggr to request that it updates its tagging database with resources related to the search.

Folksonomy Search Expansions

Furthermore, Taggr allows users to specify which search expansion method(s) (Pan, Taylor, & Thomas 2009) and which reference ontology(-ies) to use for the expansion. The extended search will first be evaluated against its tagging database. As the ontological constraints needed for the search expansions require only the expressive power of OWL 2 QL, Taggr can make use of the semantic approximation(s) of the reference OWL DL ontology(-ies) for all entailment checking. Due to the logical properties of semantic approximation, TrOWL can provide sound and complete results for all the needed entailment checking. Details of the search expansion methods go beyond the scope of this paper.

Scenario: Mashing Linked Music Data

In this section, we describe a concrete scenario illustrating how the semantic infrastructure (presented in the previous section) could enhance a typical Web 2.0 application. Let's consider the story of "Sarah", a keen Web developer with an interest in Web 2.0 and new Web technologies in general.

A Web 2.0 Application

Sarah's interest in Web 2.0 had grown from her interest in music. Meanwhile, she had been reading about Web Services and Mashup as part of her interest in Web development. After reading a few articles on the Web, Sarah decided that she could address this inconvenience by building her own mashup application. The goals of this application were to combine music-related

resources – videos, images, biographies and discographies – into a single website. Sarah also decided that if her application were to be of use, it would have to provide accurate search results in a timely fashion. Sarah named her new web application "MusicMash" and began work on the project.

Searching Folksonomies

To retrieve video and image content for her new site, Sarah made use of the public Web Service APIs provided by YouTube and Flickr. She quickly developed the first prototype. This version allowed users to search based on the artist name. MusicMash used the artist name when making calls to the YouTube and Flickr APIs to retrieve videos tagged with the each word from the artist's name.

Sarah soon found that this early version of MusicMash suffered from the major drawback that artist names are often ambiguous search terms in YouTube and Flickr. For example, when she searched for Focus (the dutch Progressive Rock band), only 5 out of the top 20 results returned by YouTube were relevant to that artist. Sarah noticed that users on YouTube often tagged music videos with both the artist name and song title. She soon realised that including a song title with the artist name resulted in much more relevant search results.

Sarah then extended MusicMash to populate a database of music metadata retrieved from the MusicBrainz¹⁰ web service. Using this metadata, Sarah could extend MusicMash to automatically expand a simple artist name search, to an *artist name plus song title* search, for each song by that artist. This search expansion technique resulted in an impressive increase in the precision of the search results. However, the volume of API calls needed for the search expansion resulted in an unacceptable amount of time for the search to return.

The Switch to Taggr

The performance issues in MusicMash were due to the large number of HTTP requests to Web Service APIs

¹⁰<http://www.musicbrainz.org/>

generated by Sarah’s search expansion technique. The solution to these issues could be addressed if the APIs could perform the search expansion internally, needing only one HTTP request. Sarah learned of a system called Taggr that provides the same search expansion functionality as MusicMash via a Web Service API. Sarah decided to redevelop MusicMash using the Taggr API, rather than accessing YouTube and Flickr directly.

The Taggr API allows Sarah to input the original search term(s) from the user and some extra parameters to specify how the search expansion should be performed. More specifically, the parameters indicate whether video or image resources should be returned; what the input search term(s) should identify, S (a music artist in this case); where to find the extra keywords for the search expansion, T (a song title in this case); and how S and T are related, P (a music artist is the creator of a song). Taggr uses OWL DL ontologies to represent its metadata internally. The parameters S and T should be specified as OWL classes and P as an OWL property. However, Sarah did not know which OWL class was used to identify music artists and song titles on the Web. She followed a link from the Taggr website to ONTOSEARCH2. She then made use of ONTOSEARCH2’s ontology search engine to find out which ontologies contained resources relating to music. Sarah typed “music” into the ONTOSEARCH2 search engine and one of the first results returned was from the Music Ontology¹¹. After further investigation Sarah found that the Music Ontology was exactly what she needed to describe the classes and properties for music artists, albums and songs.

Sarah then set about trying some searches on Taggr using the concepts `mo:MusicArtist` and `mo:Track`, related via the property `mo:foaf:made`; allowing her to replicate the search expansion from MusicMash. She first used Taggr to check which new keywords were generated by its search expansion. Sarah tried the keyword “Coldplay” and was surprised to see that Taggr did not provide any new keywords. She then searched ONTOSEARCH2 directly for “Coldplay” and again, no results were returned. Sarah realised that she would have to provide the Music Ontology individuals herself in order for the search expansion to work correctly.

From Relational Databases to Ontologies

Since ontology individuals are required by Taggr to replicate the MusicMash search expansion. Sarah decided to drop her database of music metadata in favour of Music Ontology instances stored in the ONTOSEARCH2 repository. ONTOSEARCH2’s submission and query engine provided the tools that she needed to insert new individuals and query against them. Sarah decided to populate her ontology using Web Services that can be easily linked. More specifically, using MusicBrainz API for basic artist, album and song

¹¹<http://purl.org/ontology/mo/>

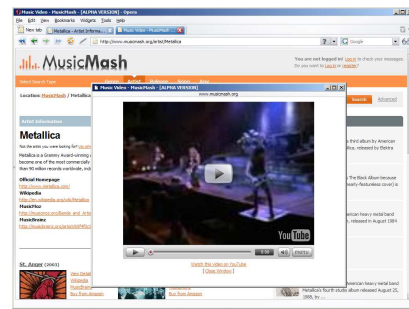


Figure 4: MusicMash2 Screenshot

information she could extend the metadata with other sources that referred to MusicBrainz identifiers, such as Last.fm and DBpedia. This new version of MusicMash was named MusicMash2.

The final two problems which were left for Sarah to address occurred on the occasions that there is no Music Ontology instances relating to a user’s search or where there were insufficient resources returned by Taggr. She decided that for any search for which MusicMash2 did not immediately return more than five results to the user, a request would be made to Taggr to populate its tagging database with more resources from tagging systems in its library. Taggr would then send requests to its supported tagging systems to retrieve the first 500 results based on the original search term(s). Similarly when MusicMash2 returns no individuals in the Music Ontology relating to the search, it initiates a background task to retrieve the required information from the Web services in its library. The advantages of this method is that information relating to previously unknown artists can be added automatically to the Music Ontology in ONTOSEARCH2 and Tagging Database in Taggr. Sarah decided that it was an acceptable trade-off that the first user should wait for the information to be retrieved, in order that future searches would return in a more acceptable amount of time.

Usefulness Evaluation

A typical scenario for MusicMash2 can be illustrated by a user searching for information related to an artist. The user first enters the name of the artist into the search box. On completion of a successful search, MusicMash2 displays information to user related to the artist. This includes a short abstract from DBpedia, the artist’s discography and links to the artist’s homepage and Wikipedia articles. The user can also select the Video Gallery tab to display videos relating to the current artist. The Video Gallery makes use of Taggr to return high precision search results for related videos. An example artist page can be viewed at the following URL: <http://www.musicmash.org/artist/Metallica>.

We can evaluate the usefulness of the two main components of our infrastructure by illustrating the ben-

efits of a Semantic Mashup such as MusicMash2 over a standard Web 2.0 application such as the original MusicMash system. Firstly, the TrOWL infrastructure provides a repository of ontologies allowing an ontology generated by one application to be easily reused in other application in the same domain. In our scenario, MusicMash2 is continually contributing to the Music Ontology stored in the TrOWL repository. In the original MusicMash RDBMS approach, all information generated by the application is locked in the application's own proprietary database and cannot be easily reused by third party applications. Secondly, Taggr's folksonomy search expansion methods provide a powerful platform for domain-specific applications to retrieve multimedia resources from tagging systems. While Taggr can be used by standard Web 2.0 applications, the information used by Taggr to perform the folksonomy search expansion is retrieved from the ontologies stored in the TrOWL repository. MusicMash2 ensures that Taggr can always find resources by keeping the music ontology up to date.

Related Work

There has been significant interests on combining Semantic Web and Web 2.0 (Benjamins *et al.* 2008; Greaves 2007; Ankolekar *et al.* 2007). In (Ankolekar *et al.* 2007) the authors present the potential for combining Web 2.0 and Semantic Web technologies in a weblog scenario, illustrating that Semantic Web and Web 2.0 are not in fact competing visions of the Web and with the right focus can be combined to overcome each other's limitations. (Greaves 2007) also discuss the combination of Web 2.0 and Semantic Web technologies, concluding that the most crucial area of Semantic Web technologies that can be of benefit to Web 2.0 lie in its query and reasoning capability. The authors of (Benjamins *et al.* 2008) believe that the integration of Semantic Web and Web 2.0 technologies can form the basis for the future generation of semantic-service based computing infrastructure.

CONCLUSION

In this paper, we have described how to use our ontology infrastructure to build semantic mashup applications like MusicMash2, which combines semantic web technologies, freely available ontologies, open sources of data, and Web 2.0 applications Flickr and YouTube. We have shown how the combination of these technologies can make folksonomy searching more accurate within a specific domain – particularly in areas where a simple keyword search is too generic to produce relevant results.

The second benefit of our approach is that by combining open ontologies with information retrieved from proprietary knowledge bases, we increase the access to this information through open interfaces. Since ONTOSEARCH2, a front end of our infrastructure, is publicly accessible through a standardised interface

(SPARQL), it is possible for other applications to be built on top of the ontologies generated by MusicMash2. By using this technique to add value to existing folksonomy based websites, we provide a carrot which may stimulate further development and/or population of domain ontologies.

Future work in this area will be focused on generalising the techniques used, putting sources of domain knowledge and folksonomy APIs into pluggable modules to encourage the development of similar tools.

References

- Ankolekar, A.; Krotzsch, M.; Tran, T.; and Vrandečić, D. 2007. The two cultures: mashing up web 2.0 and the semantic web. In *Proc. of the 16th international World Wide Web conference (WWW2007)*, 825–834.
- Benjamins, R. V.; Davies, J.; Baeza-Yates, R.; Mika, P.; Zaragoza, H.; Greaves, M.; Gomez-Perez, J. M.; Contreras, J.; Domingue, J.; and Fensel, D. 2008. Near-term prospects for semantic technologies. *IEEE Intelligent Systems* 23(1):76–88.
- Calvanese, D.; Giacomo, G. D.; Lembo, D.; Lenzerini, M.; ; and Rosati, R. 2005. DL-Lite: Tractable description logics for ontologies. In *Proc. of AAAI 2005*.
- Greaves, M. 2007. Semantic web 2.0. *IEEE Intelligent Systems* 22(2):94–96.
- Pan, J. Z., and Thomas, E. 2007. Approximating OWL-DL Ontologies. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07)*, 1434–1439.
- Pan, J. Z.; Stamou, G.; Stoilos, G.; and Thomas, E. 2007. Efficient Query Answering over Fuzzy DL-Lite Ontologies. In *Proceedings of 20th International Workshop on Description Logics (DL-2007)*.
- Pan, J. Z.; Stamou, G.; Stoilos, G.; Taylor, S.; and Thomas, E. 2008. Scalable Querying Services over Fuzzy Ontologies. In *the Proc. of the 17th International World Wide Web Conference (WWW2008)*.
- Pan, J. Z.; Taylor, S.; and Thomas, E. 2009. Reducing Ambiguity in Tagging Systems with Folksonomy Search Expansion. In *Proc. of the 6th European Semantic Web Conference 2008 (ESWC2009)*.
- Pan, J. Z.; Thomas, E.; and Sleeman, D. 2006. ONTOSEARCH2: Searching and Querying Web Ontologies. In *Proc. of WWW/Internet 2006*, 211–218.
- Passant, A. 2007. Using Ontologies to Strengthen Folksonomies and Enrich Information Retrieval in Weblogs. In *Proc. of 2007 International Conference on Weblogs and Social Media (ICWSM2007)*.
- Salton, G., and McGill, M. J. 1983. *Introduction to modern information retrieval*. McGraw-Hill.
- Thomas, E.; Pan, J. Z.; and Sleeman, D. 2007. ONTOSEARCH2: Searching Ontologies Semantically. In *Proc. of OWL: Experiences and Directions (OWLED2007)*.