A WEB OF MUSICAL INFORMATION

Yves Raimond, Mark Sandler
Centre for Digital Music
Queen Mary, University of London

ABSTRACT

We describe our recent achievements in interlinking several music-related data sources on the Semantic Web. In particular, we describe interlinked datasets dealing with Creative Commons content, editorial, encyclopedic, geographic and statistical data, along with queries they can answer and tools using their data. We describe our web services, providing an on-demand access to content-based features linked with such data sources and information pertaining to their creation (including processing steps, applied algorithms, inputs, parameters or associated developers). We also provide a tool allowing such music analysis services to be set up and scripted in a simple way.

1 INTRODUCTION

Information management has become a primary concern for multimedia-related technologies, from personal collections management to the construction of large content delivery services. However, the solutions that have emerged still exist in isolation. For example, large online databases (such as Musicbrainz, MyStrands, etc.) and personal music collection management tools such as iTunes or Songbird do not interact with each other, although they actually deal with the same kind of data. The information one of these solutions manages does not benefit from the information another may hold.

The problem becomes acute when narrowing our view to the exchange of results between music technology researchers. If providing access to content-based feature extraction through web services is a first step [10, 6], the results they produce, in order for them to be meaningful, must be interlinked with other data sources. Just giving back a set of results from a particular digital audio item is useless unless we know *what* has been processed, and *how*.

In this paper, we show the benefits of using a set of web standards, often referred to as Semantic Web technologies, to achieve these goals. First, we give an overview of these standards and how they can be used to create a 'web of data'—a distributed, domain-independent, web-scale database. We give a brief summary of the Semantic Web ontology we described in [12], able to deal with music-related data. We then focus on the music-related interlinked datasets we published since, and give examples of the type of queries they can answer and of tools consuming their data. We describe our lightweight extension of this music ontology to make content-based features part of this 'web of data'. We also describe our web services, providing on-demand access to such resources, as well as the framework underlying them.

2 TOWARDS A MUSIC-RELATED WEB OF DATA

2.1 Web identifiers and structured representations

The web is built around the concept of URI (Uniform Resource Identifier [1]). A URI can identify anything, not just a document: a person, a particular performance, a signal, a particular content-based feature, etc. Web resources can have multiple associated representations. For example, to a URI identifying a particular signal, we may want to associate an HTML representation of it (providing some textual information about its characteristics) or an image depicting the actual waveform. The web aspect comes into place when other web identifiers are mentioned within such a representation. For example, the HTML representation of our signal URI might link to an URI identifying the corresponding recording device.

Now, these representations can be *structured*: they can provide explicit machine-processable information. The Resource Description Framework (RDF [2]) allows such representations to be made, by expressing *statements* about web resources in the form of *triples*: subject, predicate and object. When such representations quote other resource identifiers, enabling access to corresponding structured representation, we create a 'web of data'.

For example, starting from the URI of a band available in our Jamendo dataset[3]: `http://dbtune.org/jamendo/artist/5`, two possible representations are available. One is in HTML and can be rendered for human consumption through a traditional web browser; the other is structured and machine readable, holding explicit statements about this band. The RDF representation, requested via the HTTP `Accept` header, is

[1] http://www.ietf.org/rfc/rfc2396.txt
[2] `http://www.w3.org/RDF/`
[3] See `http://dbtune.org/` for all our DBTune datasets.

as follows: [4]

```
<http://dbtune.org/jamendo/artist/5>
    a mo:MusicGroup;
    foaf:made <http://dbtune.org/jamendo/record/174>;
    foaf:made <http://dbtune.org/jamendo/record/33>;
    owl:sameAs <http://zitgist.com/music/artist/0781a3f3-
         645c-45d1-a84f-76b4e4decf6d>;
    foaf:based_near <http://sws.geonames.org/2991627/>;
    foaf:homepage <http://www.both-world.com>;
    foaf:img <http://img.jamendo.com/artists/b/both.jpg>;
    foaf:name "Both"^^xsd:string.
```

Using this representation, we can for example follow the `foaf:based_near` link to a resource in the Geonames dataset [5] in order to get detailed information about the place where this band is based.

2.2 Accessing and querying Semantic Web data

The new SPARQL W3C recommendation [6] gives a way to access and query such data, through a simple SQL-like syntax allowing requests ranging from simple DESCRIBEs ("return all information about resource x") to complex structured queries (eg. "return the latest album from artists located in Mexico, produced by a French person").

SPARQL allows us to specify explicitly the location of the RDF data, access a particular aggregation of data, or drive a Semantic Web "user agent", such as the Semantic Web Client Library [7], which crawls and aggregates data in order to satisfy a particular query. A place on the web accepting SPARQL queries is called an *end-point*.

2.3 The Music Ontology

The Music Ontology we described in [12] aims at providing a set of web identifiers and corresponding structured representations for an ontology (defining the main concepts and relationships) of the music domain. It is divided into three different parts, respectively dealing with editorial information (track names, people, labels, releases, etc.), production workflow information (compositions, arrangements, performances, recording, etc.) and event decomposition (eg. "the piano player was playing in that particular key at that time") The following Music Ontology example describes a set of resources involved in the description of a performance of Antonio Vivaldi's Four Seasons:

```
:vivaldi a mo:MusicArtist;
    foaf:name "Antonio Vivaldi";
    owl:sameAs <http://dbpedia.org/resource/Antonio_Vivaldi>.
:compositionevent a mo:Composition;
    mo:composer :vivaldi;
    mo:produced_work :thefourseasons;
    event:time [time:year "1723"].
:thefourseasons a mo:MusicalWork;
    dc:title "The Four Seasons";
    owl:sameAs
<http://dbpedia.org/resource/The_Four_Seasons_(Vivaldi)>;
    mo:part :spring.
:spring a mo:MusicalWork;
    dc:title "Concerto No. 1 in E major, Op. 8, RV 269, La primavera";
    mo:movement :largo.
:largo a mo:Movement;
    dc:title "Largo".
:performance a mo:Performance;
   owl:sameAs <http://dbtune.org/magnatune/performance/369>;
    mo:perfomer :american_baroque;
    mo:performance_of :largo.
:americanbaroque a mo:MusicalGroup;
    foaf:name "American Baroque";
    owl:sameAs
<http://dbtune.org/magnatune/artist/american_baroque>.
```

[4] All our RDF examples are written in RDF/Turtle: `http://www.w3.org/TeamSubmission/turtle/`. Each block corresponds to a set of statements (subject, predicate, object) about one subject. Web identifiers are either between angle brackets or in a prefix:name notation (with the namespaces defined at the end of the paper). Literals are enclosed in double quotes and can hold an explicit typing information.

[5] `http://geonames.org/`.

[6] `http://www.w3.org/TR/rdf-sparql-query/`

[7] `http://sites.wiwiss.fu-berlin.de/suhl/bizer/ng4j/semwebclient/`

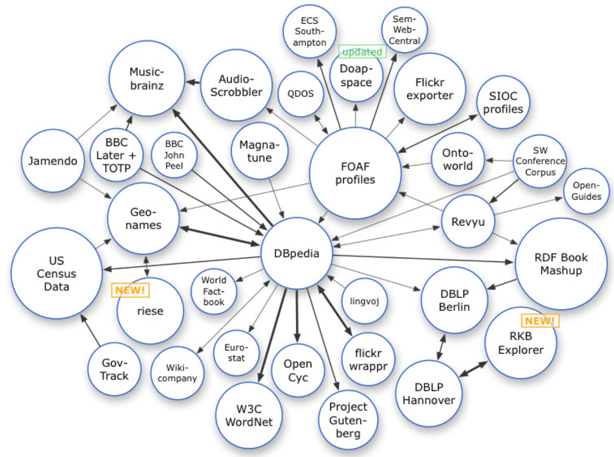

Figure 1. Available interlinked dataset, as of March 2008.

Following the principles outlined above, this example interlinks two datasets: DBpedia and Magnatune, which we study in the next section.

3 CURRENT DATASETS

The "Linking Open Data on the Semantic Web" project [3] hosted by the W3C Semantic Web Education and Outreach group aims at publishing a wide range of datasets on the Semantic Web, and creating links between them. The currently available interlinked datasets (regardless of their domain) can be depicted as in fig. 1. In this section, we discuss a few of these datasets and their links to other datasets. We especially focus on the music-related datasets we published within our DBTune service, using the Music Ontology mentioned above. We illustrate their use by examples of queries they can answer, or by tools using their data.

3.1 DBpedia

The DBpedia project described by Auer et al. in [1] extracts structured information from Wikipedia, and republishes it on the Semantic Web, together with links to other datasets. The whole DBpedia dataset holds around 100 million RDF statements, describing 2.4 million entities across a wide range of domains, and 2 million links (statements that quote an identifier outside DBpedia). For example, the following SPARQL query issued on the DBpedia dataset returns names, descriptions and birth dates of guitarists born in Paris between 1939 and 1945: [8]

```
SELECT ?name ?birth ?description ?person WHERE {
?person
  p:birthPlace <http://dbpedia.org/resource/Paris> ;
  a yago:Guitarist110151760 ;
  p:birth ?birth ;
  foaf:name ?name ;
  rdfs:comment ?description .
  FILTER (LANG(?description) = 'en' &&
   ?birth > "1939-01-01"^^xsd:date &&
   ?birth < "1945-01-01"^^xsd:date ) . }
```

The DBpedia relationship finder [9] allows this dataset to be queried for arbitrary RDF 'paths' between two different resources. For example, it would answer the query 'what do Mozart and Metallica have in common?' as depicted in fig. 2.

3.2 DBTune

We created the DBTune service to experiment with heterogeneous music-related datasets published as Music Ontology linked data. We published the Magnatune record label catalogue, and we interlinked it with corresponding DBpedia identifiers (as illustrated in § 2.3). We also published the Jamendo music platform database, and we interlinked it with Geonames and Musicbrainz identifiers. DBTune republishes on the Semantic Web data coming from AudioScrobbler and MySpace, therefore allowing different social networks as well as listening habits to be interlinked. DBTune also hosts a Semantic Web version of the BBC John Peel sessions, interlinked with DBpedia.

Musicbrainz is a community-maintained music database holding detailed editorial information about 300,000 artists, 500,000 releases and 6 million tracks. We published a linked data version of this dataset. Artists, records and countries are interlinked with corresponding DBpedia web identifiers. For example, the identifier corresponding to Madonna in DBTune is stated as being the same as the corresponding identifier in DBpedia. We also interlinked artist resources with corresponding web identifiers on our linked data publication of the MySpace social network.

We provide a number of SPARQL end-points, for each dataset (except AudioScrobbler and MySpace, as the RDF documents are generated dynamically). For example, the following query can be issued to our Jamendo end-point:

```
SELECT DISTINCT ?an ?lat ?long WHERE {
?a a mo:MusicArtist;
  foaf:based_near ?place;
  foaf:name ?an;
  foaf:made ?alb.
?alb tags:taggedWithTag
     <http://dbtune.org/jamendo/tag/jazz>.
?place
  geo:name ?name;
  geo:population ?population;
  wgs:lat ?lat;
  wgs:long ?long } ORDER BY ?population
```

This returns artists who made at least one album tagged as 'jazz' by a Jamendo user, sorted by the number of inhabitants of the places they are based near. Overall, DBTune gives access to approximately 13 billion RDF triples, thus making it one of the largest dataset on the Semantic Web.

3.3 Use-case: personal collection management

Such interlinked music-related data sources can be used for a wide range of use-cases, including personal collection management. We developed two tools to aggregate Semantic Web data describing arbitrary personal music collections. GNAT [9] finds, for all tracks available in a collection, the corresponding web identifiers in the Semantic Web publication of the Musicbrainz dataset mentioned earlier. GNAT uses primarily a metadata-based interlinking algorithm described in [13], which was also used to interlink the above described music-related datasets. GNARQL crawls the web of data from these identifiers and aggregate structured information about them, coming from heterogeneous data sources. GNAT and GNARQL then automatically create a tailored database, describing different aspects of a personal music collection. GNARQL provides a SPARQL end-point, allowing this aggregation of Semantic Web data to be queried. For example, queries such as "Create a playlist of performances of works by French composers, written between 1800 and 1850" or "Sort European hip-hop artists in my collection by murder rates in their city" can be answered using this end-point. GNARQL also gives a faceted browsing interface based on /facet [8], as illustrated in fig. 3.

4 ON-DEMAND FEATURE EXTRACTION AND INTERLINKING

4.1 Why web services are not enough

Although providing web service access through technologies such as SOAP or WSDL is a first step towards a distributed framework for performing music analysis tasks (as highlighted in [10] and [6]), this is not enough. In order for results to be easily re-usable, it is desirable to prevent

[8] This query returns, among others, Babik Reinhardt, one of Django Reinhardt's sons.

[9] GNAT and GNARQL are available at `http://sourceforge.net/projects/motools`.

Figure 2. What do Mozart and Metallica have in common?

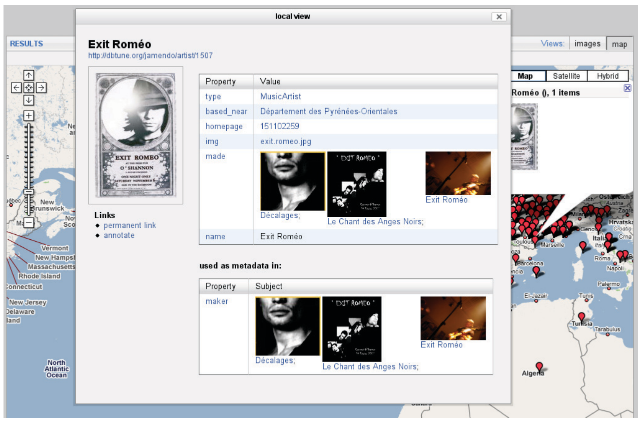

Figure 3. Management of personal music collections using GNAT and GNARQL. Here, we plot our collection on a map and display a particular artist.

the client software from having to explicitly adapt to the peculiarities of each web service. This is particularly important when trying to combine several web services (known as the ‘choreography’ problem). The results of various services must be interlinked with additional resources supplying information pertaining to their creation. The information could include processing steps, applied algorithms, inputs and the parameters, associated developers, etc.

The technologies described in § 2 provide a way to address this issue. The results of a music analysis task can be described in an RDF document which provides such links. The results are then part of the data web, and can be queried through their relationships to further data. For example, one could issue a SPARQL query such as “Give me a structural segmentation obtained using an implementation of algorithm x developed by y, on the signal identified by z”.

In the rest of this section, we describe our framework to express music analysis results in RDF, and a tool providing on-demand access to such interlinked results. We also describe a way to easily script and publish combinations of music analysis results made available on the Semantic Web.

4.2 Representing features in RDF

We extended the Music Ontology framework to describe a wide range of musical audio features in RDF [10]. Features describing the whole audio signal (whose concept is identified by the URI `mo:Signal`) can be directly attached to the signal resource. Moreover, an audio signal is associated with a temporal extent (`tl:Interval`) defined on a timeline (`tl:TimeLine`). Then, events (`event:Event`) can be defined in order to classify particular points or intervals on such time-lines. Defining new types of audio features is then just a matter of subsuming the event concept. For example, the following defines a new zero-crossing rate feature type:

```
:ZeroCrossingRate
    a owl:Class;
    rdfs:subClassOf event:Event;
    rdfs:label "Zero Crossing Rate event".
:zcr
    a owl:DatatypeProperty;
    rdfs:subPropertyOf event:literal_factor;
    rdfs:domain :ZeroCrossingRate;
    rdfs:label "Associated zero-crossing rate".
```

Then, an instance of such a feature on the time-line of a particular signal (here, a track from the Jamendo dataset) would be represented as:

```
<http://dbtune.org/audio/Den-Nostalia.ogg> mo:encodes :sig.
:sig a mo:Signal;
    mo:time      [ # Temporal extent of the signal
      tl:timeline :tl; # Time-line on which this extent is defined
      tl:duration "PT4M50S" ]. # 4 minutes and 50 seconds
:e1 a :ZeroCrossingRate;
    :zcr "27"; # Zero-crossing rate
    event:time [
      tl:start "PT0S"; # Starts at 0
      tl:end "PT0.046439912S"; # Ends at 1024/22050 seconds
      tl:timeline :tl ]. # Defined on the time-line backing our signal
```

4.3 Evaluating music analysis predicates

We model music analysis processes as RDF properties, associating their results with the inputs and parameters used. Predicates involving these properties can be considered as *built-in* predicates, within particular SPARQL end-points. For example, the decoding of an audio file and the evaluation of a musical key detection Vamp plugin [11] implementing the template-based algorithm described by Noland and Sandler in [11] could be triggered while evaluating the following query:

```
SELECT ?sig ?result
WHERE {
<http://dbtune.org/audio/Den-Nostalia.ogg>
   mo:encodes ?sig.
?sig vamp:qm-keydetector ?result }
```

The audio file is discovered at querying time. The corresponding signal resource is associated to the `?sig` variable,

[10] `http://purl.org/ontology/af/`

[11] `http://www.vamp-plugins.org/`.

and the results given by the Vamp plugin are associated to the `?result` variable. The predicates `mo:encodes` and `vamp:qm-keydetector` both correspond to a particular process, which is triggered when the predicate is evaluated.

Moreover, in the representations of `mo:encodes` and `vamp:qm-keydetector`, we can access two statements specifying that these are *functional* properties [12]. For one subject of these properties, there is only one corresponding object. Results corresponding to the evaluation of such predicates are therefore cached, to prevent repeating similar evaluations.

In these representations, we might also have links to actual implementations. We can therefore discover at querying time how to evaluate a new predicate by retrieving such implementations.

4.4 Combining music analysis and structured web data

N3 [2] provides a way to publish rules over RDF data on the Semantic Web, by extending the RDF data model in two ways: *quoting* facilities (the ability to consider an RDF graph as a literal resource) and universal quantification. A particular RDF predicate then captures the notion of implication: `log:implies` (denoted by the => symbol in our example). N3, in our context, allows to easily script how combination of predicates (which resolution might trigger some analysis tools, or some querying of RDF data) can lead to further predicates. A N3 rule can express "statements linking two similar audio signals can be derived by computing a MFCC model for both and thresholding their Kullback-Leibler divergence", or "statements linking two similar audio signals can be derived by thresholding the cosine distance of the corresponding beat spectra" [13]. A N3 rule can also merge contextual queries and content analysis, as suggested in [7]: "A *danceable* track has an average tempo of around 120 beats per minute, a high loudness, and is in the playlist of a disc jockey.".

Our previous key detection example can be mashed up with an identification process (captured by the `gnat:match` predicate which, from an audio file, finds out the corresponding identifier in the Musicbrainz dataset) and be translated into Music Ontology terms using the following rule:

```
{ ?af mo:encodes ?sig; gnat:match ?mbzsig. ?sig vamp:qm-keydetector ?results.
?mbzsig mo:time [tl:timeline _:tl ].
(?start ?duration ?key) list:in ?results. } => {
    _:keyevent a to:Tonality;
      rdfs:label "key event"; to:key ?key; event:time [
        tl:start ?start; tl:duration ?duration; tl:timeline _:tl ]. }
```

[12] individuals of `owl:FunctionalProperty`

[13] Using the approach described in [5], we can trace back what has been achieved to derive such similarity statements.

4.5 On-demand extraction and interlinking: Henry

We developed an interpreter of such rules combining music analysis processes and contextual data: Henry [14]. Henry handles two persistent stores: an RDF and N3 store m (handling, among other things, computation metadata and RDF statements gleaned from the web) and a binary store b (handling binary representations of resources described in the RDF store: signals, features, etc.). The underlying logic of this software builds on top of Transaction Logic [4] over b. Henry provides the ability to register new built-in predicates written in a variety of languages.The main interface to interact with Henry is its SPARQL end-point. When processing a SPARQL query q, Henry executes the following steps:

1. For every newly appearing web identifier i in q, dereference it, and then:
   (a) If the representation is RDF, store it in m. If i is a property and its representation links to a built-in implementation matching the current platform, retrieve it ;
   (b) If the representation is N3, store it in m and register the corresponding rules ;
   (c) If the representation is binary, cache it in b ;
2. For every triple pattern p in q, the possible solutions are:
   (a) Instantiations of p in m ;
   (b) If p is in the conclusion part of a rule, the solutions correspond to the possible evaluations of the premises ;
   (c) if p=(s_p, p_p, o_p) where p_p is a built-in predicate, solutions are derived using this built-in, updating b accordingly.

4.6 Available services

An available Henry instance [15] provides access to a number of such results. It wraps a Vamp plug-in host, so new built-in predicates and the corresponding Vamp implementation can be discovered at querying time. This instance also discovers new audio files at querying time. It provides a SPARQL end-point, although a future version will also include a simplified HTTP interface.

The above defined rule for key events and a number of others are pre-loaded. For example, the following query selects several attributes of key events on the time-line of an audio signal (start time, end time and corresponding key):

[14] Henry source code is available at `http://code.google.com/p/km-rdf/`

[15] Available at `http://dbtune.org/henry/`

```
SELECT ?start ?end ?key WHERE {
<http://dbtune.org/audio/Den-Nostalia.ogg>
     mo:encodes ?sig.
?sig mo:time ?time.
?time tl:timeline ?tl.
_:evt a to:Tonality;
     event:time [
        tl:start ?start; tl:end ?end; tl:timeline ?tl
        ] ;
     to:key ?key }
```

This query triggers a call to a key detection Vamp plugin if the results have not yet previously been computed for the mentioned audio file. As the predicate corresponding to the Vamp call is functional, the Vamp results are cached. The results bound with the variable $?key$ are resources within the Tonality Ontology [16], which allows us to access further information about the keys themselves.

5 CONCLUSION AND FUTURE WORK

Publishing web resources and corresponding RDF representations providing links to further resources is a big step towards a shared information space for music enthusiasts. The web environment allows us to link together heterogeneous types of information: from music analysis results to editorial or social data, but also data that we often don't include in a music information system, such as geographical, statistical or encyclopedic data. This web of data is now a reality, which is concretely used in a number of applications such as GNAT and GNARQL. We developed a tool called Henry, which dynamically creates and publishes such linked data, by working on top of on-the-fly discovery of content, content analysis processes, logical rules, and external data. Publishing a new range of music analysis results on the data web is therefore as simple as publishing a small amount of RDF specifying new processing tools (if we need any), and some N3 rules combining them together. We are now in the position of answering hybrid queries such as 'find me all works composed in New York in 1977, performed at an average tempo of 130 bpm and with an overall C major key'.

Future work includes publishing and interlinking more music-related datasets. It also includes delegating the resolution of predicates within Henry: it might be inefficient to allow every instance to be dynamically patched with the same set of plugins, handling a similar range of processing. Moreover, some plugin providers may not want to distribute their implementation, but just provide web access to it.

6 ACKNOWLEDGEMENTS

The authors acknowledge the support of both the Centre For Digital Music and the Department of Computer Science at Queen Mary University of London for the studentship for Yves Raimond. This work has been partially supported by the EPSRC-funded ICT project OMRAS-2 (EP/E017614/1).

[16] http://purl.org/ontology/tonality/

Namespaces

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
@prefix owl: <http://www.w3.org/2002/07/owl#>.
@prefix geo: <http://www.geonames.org/ontology#>.
@prefix wgs: <http://www.w3.org/2003/01/geo/wgs84_pos#>.
@prefix yago: <http://dbpedia.org/class/yago/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.
@prefix p: <http://dbpedia.org/property/>.
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix mo: <http://purl.org/ontology/mo/>.
@prefix event: <http://purl.org/NET/c4dm/event.owl#>.
@prefix tl: <http://purl.org/NET/c4dm/timeline.owl#>.
@prefix to: <http://purl.org/ontology/tonality/>.
@prefix vamp: <http://purl.org/ontology/vamp/>.
@prefix list: <http://www.w3.org/2000/10/swap/list#>.
@prefix gnat: <http://motools.sourceforge.net/doap.rdf#>.
@prefix tags: <http://www.holygoat.co.uk/owl/redwood/0.1/tags/>.
```

7 REFERENCES

[1] S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *Proceedings of the International Semantic Web Conference*, Busan, Korea, November 11-15 2007.

[2] Tim Berners-Lee, Dan Connolly, Lalana Kagal, Yosi Scharf, and Jim Hendler. N3Logic : A logical framework for the world wide web. *Theory and Practice of Logic Programming*, 2007. To appear in Theory and Practice of Logic Programming (TPLP). Available at http://arxiv.org/abs/0711.1533. Last accessed September 2007.

[3] Chris Bizer, Tom Heath, Danny Ayers, and Yves Raimond. Interlinking open data on the web. In *Demonstrations Track, 4th European Semantic Web Conference, Innsbruck, Austria*, 2007.

[4] Anthony J. Bonner and Michael Kifer. An overview of transaction logic. *Theoretical Computer Science*, 133:205–265, 1994.

[5] Jeremy J. Carroll, Christian Bizer, Pat Hayes, and Patrick Stickler. Named graphs. *Journal of Web Semantics*, 2005.

[6] Andreas F. Ehmann, J. Stephen Downie, and M. Cameron Jones. The music information retrieval evaluation exchange "Do-It-Yourself" web service. In *Proceedings of the International Conference on Music Information Retrieval*, 2007.

[7] Roberto Garcia and Òscar Celma. Semantic integration and retrieval of multimedia metadata. In *Proceedings of the 5th International Workshop on Knowledge Markup and Semantic Annotation*, 2005.

[8] Michiel Hildebrand, Jacco van Ossenbruggen, and Lynda Hardman. *The Semantic Web - ISWC 2006*, volume 4273/2006 of *Lecture Notes in Computer Science*, chapter /facet: A Browser for Heterogeneous Semantic Web Repositories, pages 272–285. Springer Berlin / Heidelberg, 2006.

[9] Jens Lehmann, Jrg Schppel, and Sren Auer. Discovering unknown connections - the dbpedia relationship finder. In *Proceedings of the First Conference on Social Semantic Web (CSSW)*, pages 99–110, 2007.

[10] Daniel McEnnis, Cory McKay, and Ichiro Fujinaga. Overview of OMEN. In *Proceedings of the International Conference on Music Information Retrieval*, 2006.

[11] K. Noland and M. Sandler. Signal processing parameters for tonality estimation. In *Proceedings of AES 122nd Convention*, Vienna, 2007.

[12] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The music ontology. In *Proceedings of the International Conference on Music Information Retrieval*, pages 417–422, September 2007.

[13] Yves Raimond, Christopher Sutton, and Mark Sandler. Automatic interlinking of music datasets on the semantic web. In *Proceedings of the Linked Data on the Web workshop, colocated with the World-Wide-Web Conference*, 2008. Available at http://events.linkeddata.org/ldow2008/papers/18-raimond-sutton-automatic-interlinking.pdf. Last accessed June 2008.