Music Retrieval Portal

Yuyin Sun, Erjia Yan

School of Library and Information Science, Indiana University

1 Introduction

The current Web is connected via unified hyperlinks. These links cannot convey any meaning about the two objects connected. The Semantic Web, however, adds logic to these links so that each pair of links can express varied information. It can greatly enhance the capability of the information retrieval effect on the Web. The linked Data is essential for the fulfillment of Semantic Web. The data on the Web are separated, meaning that the data within one website can hardly communication those from other websites. Linked Data can facilitate the communication and interaction between related data from different data source by lowering the barriers to linking data. Wikipedia (2009) defines Linked Data as "a term used to describe a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF."

Instead of following links between HTML pages, Linked Data enable users to navigate between different data sources by following RDF links (Bizer et al., 2008). An RDF link states the relationship between two pieces of data. Linked Data is supported by the W3C Semantic Web Education and Outreach Working Group. The aim of the project is “to extend the Web with a data commons by publishing various open data sets as RDF on the Web and by setting RDF links between data items from different data sources” (SWEO, 2009). There are already various open Linked Data available: for example Wikipedia, Wikibooks, Geonames, MusicBrainz, WordNet, the DBLP bibliography and many more which are published under Creative Commons or Talis licenses. Collectively, the data sets consist of over 13.1 billion RDF triples which are interlinked by around 142 million RDF links (SWEO, 2009).

The present study is the extension of the EASAIER project (Enabling Access to Sound Archives through Integration, Enrichment and Retrieval). EASAIER seeks to develop an innovative remote access system which extends beyond standard content management and retrieval systems (EASAIER, 2008). We apply seven different Linked Data sources to the present study: Jamendo, Musicbrainz, MySpace, BBC playcount, Henry, Magnatune, and DBpedia. We build a music portal to query the RDF triples of these Linked Data through SPARQL. The rest article is
organized as follows: Section 2 introduces the data applied; Section 3 shows the interface and algorithm; Section 4 draws the conclusion and directs future research.

2 Data in the study

We apply seven Linked Data sources to the study. Table 1 shows the attributes of these sources.

- **Jamendo**: Jamendo is a large repository of Creative Commons licensed music, based in France. This server aims at publishing this dataset as Linked Data: publishing a set of URIs with an RDF representation holding links towards external datasets[^8].
- **Musicbrainz data**: MusicBrainz is a community music metadatabase that attempts to create a comprehensive music information site[^9].
- **MySpace dataset**: This service provides a live RDF representation of Myspace users. If the user is also an artist, then the corresponding tracks in the streaming audio cache are included in the RDF[^10].
- **BBC playcount data**: This service provides RDF links using these playcounts, to link the Musicbrainz linked data and the BBC programmes linked data[^11].
- **Henry**: Henry provides the ability to easily register new audio processing built-in predicates[^12].
- **Magnatune**: Magnatune is an independent music label, allowing people to buy records for as much as they want[^13].
- **DBpedia**: DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web[^14].

Table 1. Descriptions of each dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Triples</th>
<th>Description</th>
<th>URL and URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jamendo</td>
<td>1,100,000</td>
<td>A sparql endpoint for jamendo.com covering data about music artists and releases in this Creative Commons label</td>
<td>Human readable splash page: <a href="http://dbtune.org/jamendo/">http://dbtune.org/jamendo/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPARQL URI and interface: <a href="http://dbtune.org/jamendo/sparql/">http://dbtune.org/jamendo/sparql/</a></td>
</tr>
<tr>
<td>Musicbrainz</td>
<td>36,000,000</td>
<td>powered by D2R server and a D2RQ mapping, SPARQL end-point available</td>
<td>Human readable splash page: <a href="http://dbtune.org/musicbrainz/">http://dbtune.org/musicbrainz/</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>SPARQL URI and interface: <a href="http://dbtune.org/musicbrainz/sparql">http://dbtune.org/musicbrainz/sparql</a></td>
</tr>
</tbody>
</table>

[^8]: http://dbtune.org/jamendo/
[^9]: http://musicbrainz.org/
[^10]: http://dbtune.org/myspace/
[^11]: http://dbtune.org/bbc/playcount/
[^12]: http://dbtune.org/henry/
[^13]: http://dbtune.org/magnatune/
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Source</th>
<th>Description</th>
<th>Human readable splash page:</th>
<th>SPARQL URI and interface:</th>
</tr>
</thead>
<tbody>
<tr>
<td>MySpace</td>
<td>N/A</td>
<td>A sparql endpoint containing some data about Myspace profiles</td>
<td><a href="http://dbtune.org/myspace/">http://dbtune.org/myspace/</a></td>
<td><a href="http://virtuoso.dbtune.org/sparql">http://virtuoso.dbtune.org/sparql</a></td>
</tr>
<tr>
<td>BBC playcount</td>
<td>1,954,786</td>
<td>linking The BBC Programmes linked data and the Musicbrainz linked data, SPARQL end-point available</td>
<td><a href="http://dbtune.org/bbc/playcount/">http://dbtune.org/bbc/playcount/</a></td>
<td><a href="http://dbtune.org:3062/sparql">http://dbtune.org:3062/sparql</a></td>
</tr>
<tr>
<td>Henry</td>
<td>N/A</td>
<td>A SPARQL end-point interpreting signal processing workflows and providing on-demand content-based data</td>
<td><a href="http://dbtune.org/henry/">http://dbtune.org/henry/</a></td>
<td><a href="http://dbtune.org/henry/sparql/">http://dbtune.org/henry/sparql/</a></td>
</tr>
<tr>
<td>Magnatune</td>
<td>N/A</td>
<td>A sparql endpoint for magnatune.com covering data about music artists and their releases on the Magnatune label</td>
<td><a href="http://discogs.dataincubator.org/.html">http://discogs.dataincubator.org/.html</a></td>
<td><a href="http://dbtune.org/magnatune/sparql/">http://dbtune.org/magnatune/sparql/</a></td>
</tr>
<tr>
<td>DBpedia</td>
<td>274 million</td>
<td>Using three classifications (Wikipedia Categories, YAGO Classification, and Word Net Synset Links) within SPARQL queries</td>
<td><a href="http://dbpedia.org/">http://dbpedia.org/</a></td>
<td><a href="http://dbpedia.org/sparql">http://dbpedia.org/sparql</a></td>
</tr>
</tbody>
</table>

### 3 Interface and Sample Algorithms

The music retrieval portal is implemented in the Struts framework, which mainly focuses on request dispatching. We consider the Module-View-Control (Figure 1) model as the main structure. Figure 1 shows the structure of the system. In facts, there is no exact module layer in the system, since we do not need any persistence techniques to support the system. However, we create a modular Data Access Object (DAO) to maintain the SPARQL query process. Meanwhile, a service interface, which manages the query service, is proposed in the system. It defines the basic service function of the control layer. We implement the result processing modules of different data sources based on this interface. We also use many utilities in the results processing part, such as HTML filter and HTML code wrapper. In the view layer, we adopt action-dispatching process of Struts. It manages the search request and works as an interceptor for the potential purpose. The current system, we do not implement the request and parameter processing function, but simply dispatch the request to a certain service module. When display results from different data sources, we employ Ajax to retrieve the asynchronize data from different actions. Figure 3 shows the search interface and Figure 4 is result display interface.
Figure 1. The search process mechanism of the Music Retrieval Portal.

Figure 2. The structure of the Music Retrieval Portal.
Figure 3. The search interface of the Music Retrieval Portal

Figure 4. The result interface of the Music Retrieval Portal
The SPARQL sample queries of different data sources are listed below:

**DBPedia:**


**Jamendo:**


**Magnatune:**

MusicBrainz:


The present study shows an example of how to integrate and query different sources of Linked Data through SPARQL. The advantage is that users can search in the music portal and obtain the results from different sources. The results may cover the queried artist from different aspect, which are more informative and potentially more relevant. Meanwhile, the results are in the consistent format, indicating that users will spend less time in understanding and extract the information. Another advantage is that no database is needed to conduct the query. In traditional relational database, the query must be based on the data within the database, and of course, one can only query in the particular database. In Linked Data, data are store at different locations, and through SPARQL URI, we are able to query these spread data sources in a single portal without downloading and centralizing them.

We intend to incorporate more data sources, to add more search functions, including genre, country, publisher, and so on, and also to build a music ontology that have rich reasoning competence. It will allow us to make full use of the Semantic Web technology to create a large machine understandable cultural environment (Raimond et al., 2007).

Reference


