

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/265299863>

Linked Music Data from Global Music Charts

Conference Paper · September 2014

DOI: 10.1145/2660517.2660536

CITATIONS

2

READS

724

4 authors:



Milos Jovanovik

Ss. Cyril and Methodius University in Skopje

58 PUBLICATIONS 184 CITATIONS

[SEE PROFILE](#)



Matej Petrov

Ss. Cyril and Methodius University in Skopje

1 PUBLICATION 2 CITATIONS

[SEE PROFILE](#)



Bojan Najdenov

Ss. Cyril and Methodius University in Skopje

8 PUBLICATIONS 38 CITATIONS

[SEE PROFILE](#)



Dimitar Trajanov

Ss. Cyril and Methodius University in Skopje

151 PUBLICATIONS 537 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



SAGE: Semantic Geospatial Analytics [View project](#)



F-NLP: Natural Language Processing in Finance [View project](#)

Linked Music Data from Global Music Charts

Milos Jovanovik

Faculty of Computer Science
and Engineering

Skopje, Macedonia

milos.jovanovik@finki.ukim.mk

Matej Petrov

Faculty of Computer Science and
Engineering

Skopje, Macedonia

petrov_matej@yahoo.com

Bojan Najdenov

Faculty of Computer Science and
Engineering

Skopje, Macedonia

bojan.najdenov@finki.ukim.mk

Dimitar Trajanov

Faculty of Computer Science
and Engineering

Skopje, Macedonia

dimitar.trajanov@finki.ukim.mk

ABSTRACT

Accessing data on the Web in order to obtain useful information has been a challenge in the past decade. The technologies of the Semantic Web have enabled the creation of the Linked Data Cloud, as a concrete materialization of the idea to transform the Web from a web of documents into a web of data. The Linked Data concept has introduced new ways of publishing, interlinking and using data from various distributed data sources, over the existing Web infrastructure. On the other hand, music represents a big part of the everyday life for many people in the world, and therefore, understandably, the Web contains loads of data from the music domain. Given the fact that Linked Data enables new, advanced use-case scenarios, the music domain and its users can also benefit from this new data concept. Besides being provided with additional information about their favorite artists and songs, the users can also potentially get an overview of the dynamics of the global music playlists and charts, from the aspects of artists, countries, genres, etc. In this paper, we describe the process of transforming one- and two-star music playlists and charts data from various global radio stations, into five-star Linked Data, in order to demonstrate these benefits. We also present the design of our Playlist Ontology necessary for our data model. We then demonstrate – via SPARQL queries and a web application – some of the new use-case scenarios for the users over the published linked dataset, which are otherwise not available over the isolated datasets on the Web.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SEM '14, September 04 - 05 2014, Leipzig, AA, Germany
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2927-9/14/09\$15.00.
<http://dx.doi.org/10.1145/2660517.2660536>

Categories and Subject Descriptors

H.3.5 [Information storage and retrieval]: On-line information Services – *Data sharing, Web-based services*; H.2.4 [Database Management]: Systems – *Distributed databases*.

General Terms

Algorithms, Design, Experimentation.

Keywords

Music, Linked Data, Open Data, Playlist Ontology.

1. INTRODUCTION

Technology has always been a major tool in improving the quality of life for people. By lowering the barrier for publishing and accessing documents, the Web has been the innovation which changed the way we communicate, as well as the way we gather and share knowledge. However, the original design of the Web has been intended for human consumption only, so in order to obtain and analyze larger amounts of data, intelligent software tools are needed. The technologies of the Semantic Web represent a set of standards which can be applied over the documents on the Web, or any other data, in order to enable interlinking of the different data sources into a web of data. This lowers the data access barrier even further for simpler software tools to be able to obtain, understand, process and use it [1][2][3].

The concept of Linked Data represents a concrete materialization of the Semantic Web vision. It is a set of best practices which can be used for publishing, interlinking and querying data from different and distributed data sources, over the existing infrastructure of the Web. As part of the Linked Data endeavor, the Linking Open Data (LOD) Cloud¹ has been created. It consists of a large number of interlinked datasets, from different domains, which have been published on the Web. With this, the LOD Cloud represents a rich network of data, which can be accessed using the technologies of the Semantic Web.

The Linked Data concept and the LOD Cloud allow the creation of use-case scenarios for both users and their applications which have not been available before, in isolated datasets. Therefore, they can be used in new, innovative applications in various

¹ <http://lod-cloud.net/>

domains, which would leverage the value of the data, and create new business value in the industries [4][5].

For the purpose of measuring data quality on the Web, Tim Berners-Lee has proposed a 5-star rating system². According to the rating system, each information published online gets at least one star. Data published in machine-readable, structured formats get two stars, and data published in non-proprietary structured formats (CSV, XML, etc.) get three stars. Four stars are given to data which use Semantic Web standards (RDF, OWL, SPARQL, etc.) for structure and access, and five stars are reserved for data which additionally link to other people's data, for providing context.

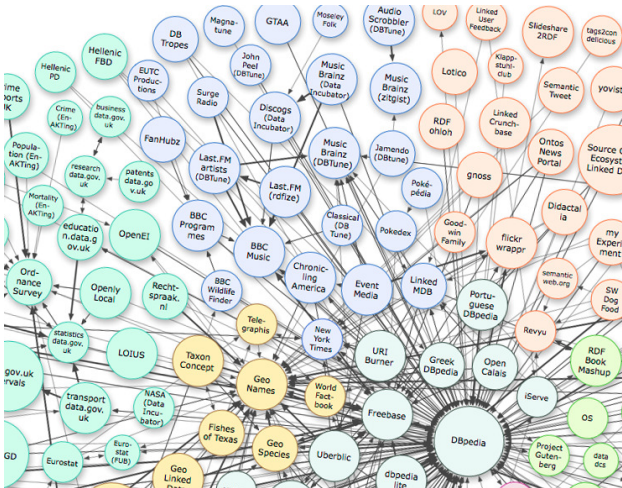


Figure 1. The media part of the LOD Cloud³.

In order to create new, advanced use-case scenarios for the music audience, we need to apply the principles of Linked Data over existing music related data on the Web. One particular domain where these new data publishing and access principles can help is the domain of music data from global playlists and charts; these data represent the music taste of the general public, and provide a snapshot of which artist, song or genre was globally popular in a particular moment in time. Having these data as Linked Data can provide the general users with more information about their favorite artist, songs, genres, but also allow them to get an overview of the dynamics of the global music playlists and charts, from various aspects. This can be achieved through applications which access the Linked Data available on the Web.

In this paper, we present a sustainable system and its methodology for obtaining and transforming one-star and two-star music related data from the websites of various global radio stations, into five-star Linked Data, interlinked with music domain datasets from the LOD Cloud (Figure 1). We also provide example use-case scenarios over the created dataset – via SPARQL queries and a web application – which demonstrate the advantage of interlinked over isolated datasets.

The paper is organized as follows: in SECTION 2 we discuss related work and present music related datasets which are part of the

LOD Cloud. In Section 3 we describe the design of the process of gathering and staging data, and their transformation and interlinking using the Linked Data practices. Here, we also discuss our data model, the reuse of existing ontologies and our Playlist Ontology, which we designed specifically for this purpose. In Section 4 we present and demonstrate example use-cases which arise from the interlinked datasets. In Section 5, we present our web application built on top of the dataset, and in Section 6 we give a conclusion to the presented work.

2. RELATED WORK

As we already mentioned, music related datasets are already part of the LOD Cloud (Figure 1). They have been created by various projects, and we will take a closer look at them.

DBTune⁴ is a project aimed at providing access to music related data, published following the Linked Data principles. It provides access to more than 14 billion RDF Triples from various datasets, such as MySpace, Jamendo, Last.fm, MusicBrainz, etc. The datasets which are part of the DBTune project are represented as blue circles on Figure 2. Both Figure 1 and Figure 2 show the connections the DBTune datasets have with other datasets from the LOD Cloud.

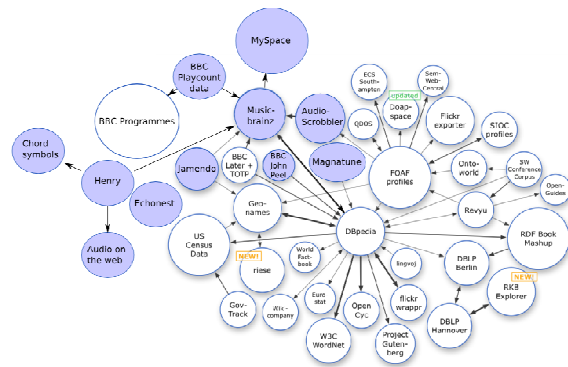


Figure 2. DBTune Datasets, depicted as blue circles.

MusicBrainz⁵ represents an open source repository of music information, which is community-maintained. The data stored in the MusicBrainz database is very diverse, spanning from data about artists and their releases and albums, to publishers, composers, etc. Although it is publicly available and free to use, MusicBrainz does not serve its data in a Linked Data manner directly. Regardless, since it provides unique identifiers for artists, albums and tracks, it is already widely used as a source for music-related URIs in the LOD Cloud.

LinkedBrainz⁶ is a project which is intended to publish the MusicBrainz data as Linked Data. As a result of the project, the MusicBrainz data is exposed in RDF using mappings of concepts from its database into concepts of the Music Ontology⁷ and other appropriate ontologies. The project also provides dereferenceable URIs for the entities and a public SPARQL endpoint for querying the MusicBrainz data.

² <http://5stardata.info/>

³ Taken from the Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. <http://lod-cloud.net/>

⁴ <http://dbtune.org/>

⁵ <http://musicbrainz.org/>

⁶ <http://linkedbrainz.org/>

⁷ <http://musicontology.com/>

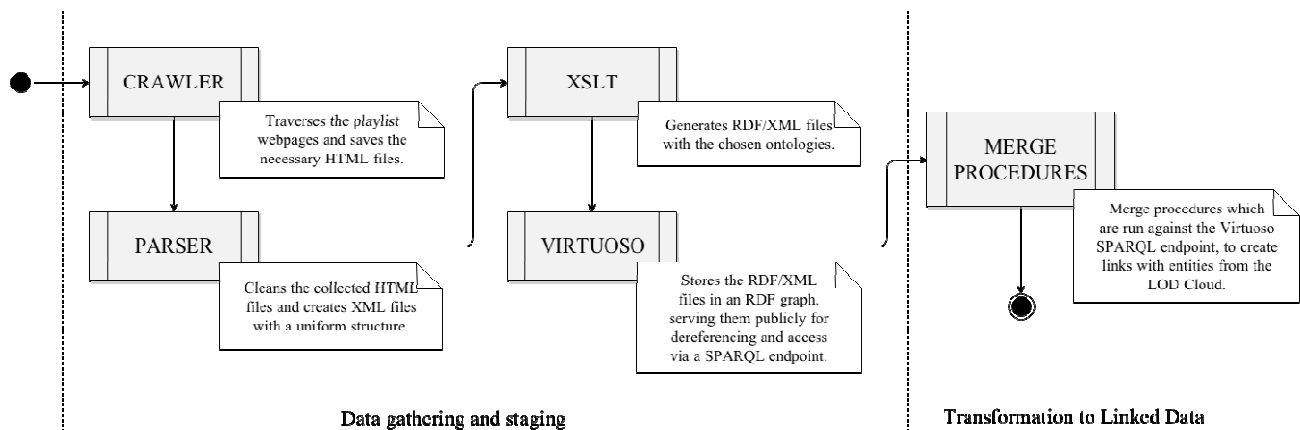


Figure 3. The workflow of obtaining, transforming, publishing and interlinking the data.

Another notable project is the music recommendation system, based on social networking and user contribution [6]. The goal of this system is to provide means to use interlinked data from the LOD Cloud and combine them with social and user data, in order to provide data-rich recommendations.

3. GENERATING LINKED DATA FROM GLOBAL RADIO STATIONS

Although we have already worked on generating Linked Data in other domains [7][8][9][10][11], we had no previous experience with music data. After we did our research on available music data to support our research idea, we decided to use the public data from the official music playlists and charts from various global radio stations, which are published on their websites. Since the generator of these playlists is the listener, i.e. the Web user, we believe that providing him/her with additional use-case scenarios for information retrieval while browsing his/her favorite artists, songs and releases, can be a potential source for application development.

In order to provide these scenarios, we created a system which can obtain, transform, publish, interlink and update the playlist data. It consists of several parts which constitute one automated workflow; the workflow can then be scheduled, in order to update the data on a regular basis.

The radio stations we use as data sources are radio stations from the BBC website⁸: Radio 1, Radio 1Xtra, Radio 2, Radio 6 Music, Asian Network, Radio Scotland; the official charts from BBC Radio 1: the Official UK Top 40 Singles Chart, Dance Singles, Indie Singles, Rock Singles, etc. We chose these sources based of the amount and type of data they contain. The information about these playlists, though represented differently on each radio station website, in a non-uniform fashion, generally contains the name of the song, its current position in the playlist and the name of the artist performing it.

The playlists are available only as HTML tables on the radio station websites. Therefore, we use a custom crawler to obtain and clean the playlist data, and to store it locally in XML format. After that we transform them from XML to RDF/XML format, load

them into an RDF Graph and link them with datasets from the LOD Cloud.

The automated workflow of generating Linked Data from the playlists goes as follows (Figure 3):

1. Data gathering and staging
 - a. We use a custom web crawler to crawl and gather the HTML pages of the playlists and charts of interest.
 - b. A parser is used for cleaning and filtering the data from the HTML pages, before storing them as XML files with a uniform structure.
 - c. An XSL transformation is applied over the filtered XML content, in order to generate RDF/XML files with annotated content.
 - d. The RDF/XML files are then loaded in a Virtuoso instance, into an RDF graph.
2. Transformation to Linked Data
 - a. SPARQL-based merge procedures are run over the data from the RDF graph, in order to create links to existing entities in the LOD Cloud, i.e. generate Linked Data.

Each of these steps is depicted on Figure 3 and is described in more details below.

3.1 Data Gathering and Staging

The process of data gathering is done with a custom web crawler, which stores the HTML pages locally. Since the HTML structure of these pages varies significantly, we use a parser to extract the necessary data (playlist name, radio station name, list of songs in the playlist with their corresponding position in the playlists, etc.) from each of them. With this, we get cleaned HTML files, which we store locally as XML files.

The stored XML files are then put through an XSL transformation process, which outputs RDF data, in RDF/XML format. Even though RDF/XML has been out of favor in the Linked Data and the Semantic Web community because of its verbose syntax, it is quite convenient for use with XSL transformations, since it can be generated directly. The XSL transformation uses the scheme described further in 3.1.1 for transforming the XML elements and attributes into RDF triples in an RDF/XML format.

⁸ <http://www.bbc.co.uk/radio/>

The RDF/XML files are loaded into a Virtuoso Universal Server⁹ instance, into a single RDF graph. Each time the automated workflow is being run, it adds data into the same RDF graph, i.e. it updates the dataset. This RDF graph¹⁰ has been published and is available via a persistent URL. Its content is dereferenceable via HTTP content negotiation, as well.

3.1.1 Playlist Ontology

In order to transform the playlist data from HTML to RDF/XML, using XSL transformation, we need an ontology. As we previously mentioned, LinkedBrainz is a project which publishes the MusicBrainz data in RDF format, by using mappings of concepts from the database with concepts defined in the Music Ontology. The Music Ontology is used as a vocabulary for describing a wide range of music related information. It provides classes and concepts such as artists, albums, tracks and properties such as biography, duration, instrument and many others [12].

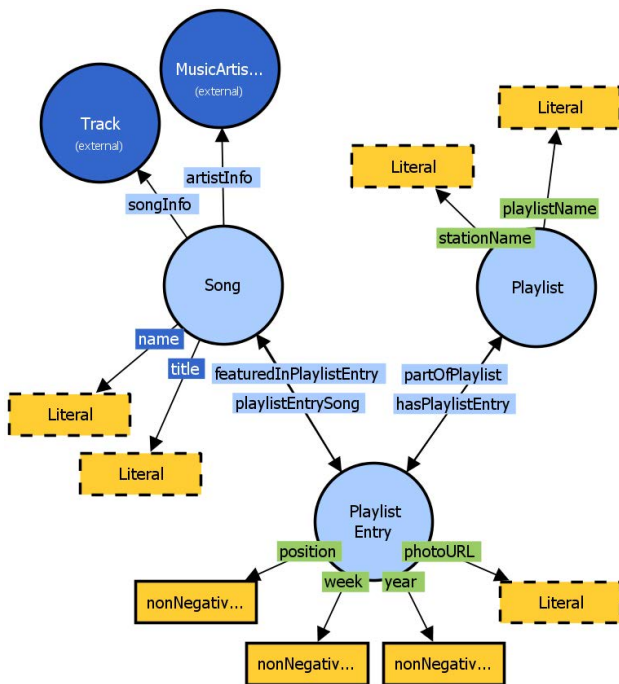


Figure 4. Diagram of the Playlist Ontology.

However, since the entities described in our dataset are playlist entries which have a different schema from the entities from MusicBrainz, we are unable to use the classes and properties from the Music Ontology for annotation purposes. Therefore, we created our own ontology, the Playlist Ontology¹¹. It is comprised of classes and properties which are necessary for describing the data from our playlist dataset. In order to support the interlinking of the data from our dataset with data from the LOD Cloud, we also needed object properties in the ontology which would serve as links between entities from the different datasets.

The Playlist Ontology has three classes (Figure 4). The po:PlaylistEntry class is used for representing an entry from a playlist. This entry is not simply a song, but rather a song which holds a specific position in a specific playlist, at a specific time. The po:Playlist class is used for representing a playlist from a radio station, and the po:Song class is used for representing a song.

Table 1. Object Properties of the Playlist Ontology.

Property	Description
hasPlaylistEntry	Used for linking a po:Playlist instance with po:PlaylistEntry instances, for entries which are part of the playlist. An inverse property of po:partOfPlaylist.
partOfPlaylist	Used for linking a po:PlaylistEntry instance with a po:Playlist instance. An inverse property of po:hasPlaylistEntry.
playlistEntrySong	Used for linking a po:PlaylistEntry instance with a po:Song instance. An inverse property of po:featuredInPlaylistEntry.
featuredInPlaylistEntry	Used for linking a po:Song instance with a po:PlaylistEntry instance. An inverse property of po:playlistEntrySong.
artistInfo	Used for linking a po:Song instance with an mo:MusicArtist instance from the LOD Cloud.
songInfo	Used for linking a po:Song instance with an mo:Track instance from the LOD Cloud.

Table 2. Datatype Properties of the Playlist Ontology.

Property	Description
position	Position of the entry in the playlist, for the specific week and year.
week	The week of the occurrence of the entry in the playlist.
year	The year of the occurrence of the entry in the playlist.
photoURL	A URL to a photo for the entry.
playlistName	The name of the playlist.
stationName	The name of the radio station.

These classes are interconnected with four object properties (Table 1, Figure 4); po:hasPlaylistEntry and po:playlistEntrySong are the main properties of the model, and po:partOfPlaylist and po:featuredInPlaylistEntry are their inverse properties, respectively. Even though having inverse properties generally introduces redundancy, i.e. writing more triples for the same information, we defined them for better SPARQL query performance for some of the use-cases. The po:Song class also uses two other object properties for connecting with LOD

⁹ <http://virtuoso.openlinksw.com/>

¹⁰ <http://purl.org/net/lmd/data>

¹¹ <http://purl.org/net/po#>

instances annotated with the Music Ontology (Table 1, Figure 4). The ontology also contains six datatype properties (Table 2, Figure 4).

In addition to these properties, we also used the foaf:name property from the FOAF Ontology¹², in order to define the name of the artist who performs the po:Song instance, and the dc:title property of the DCMI ontology¹³, in order to define the title of a po:Song instance (Table 3, Figure 4).

Table 3. Other Datatype Properties used in our Model.

Property	Description
foaf:name	Used for the artist name of a po:Song instance.
dc:title	Used for the song title of a po:Song instance.

The Playlist Ontology has been published following the best practices¹⁴, i.e. with a persistent URI, and is dereferenceable via HTTP content negotiation.

3.2 Transformation to Linked Data

After the RDF graph is created, we need to transform its data into Linked Data. In order to do this, we need to link the data from our graph to data from other datasets in the LOD Cloud. The external dataset we chose for interconnecting was MusicBrainz, or more specifically LinkedBrainz, since it contains the same data in RDF format and is accessible via a SPARQL endpoint.

In order to accomplish this, we use the two object properties from our ontology, po:songInfo and po:artistInfo. We use the po:songInfo property to connect a po:Song instance to the mo:Track instance described on LinkedBrainz. To do this, we search for an mo:Track instance on LinkedBrainz which has the same song title as our po:Song instance and is performed by the same artist, and add it as an object in an RDF triple which connects the po:Song instance with it, via the po:songInfo property (Figure 4).

In a similar manner, we use the po:artistInfo property to connect a po:Song instance to an mo:MusicArtist instance from LinkedBrainz, where the matching is done by the name of the artist performing the po:Song and the mo:MusicArtist name.

This logic was implemented in merge procedures via SPARQL queries, which are triggered from a script after the RDF graph is created or updated (Figure 3).

These po:songInfo and po:artistInfo relations represent a gateway into more details about the song and artist in question, and enable a large number of new use-case scenarios. After establishing these links between our playlist dataset and the LinkedBrainz dataset, we are able to access more data and retrieve more information about the song and the artist not only from this dataset, but also from all other LOD Cloud datasets which are interconnected with it (Figure 1). This allows us to potentially traverse the entire LOD Cloud, by starting from our dataset and playlist entries, which adds to the number of potential uses of the playlist dataset.

¹² <http://xmlns.com/foaf/spec/>

¹³ <http://purl.org/dc/elements/1.1/>

¹⁴ <http://www.w3.org/TR/ld-bp/>

4. USE CASES

As we already pointed out, our goal is to demonstrate that the transformation of playlist data into Linked Data can provide new use-case scenarios for the domain users and their applications. The technologies of the Semantic Web allow data retrieval over a distributed environment, via SPARQL federation. We will use this feature, which allows execution of SPARQL queries over distributed SPARQL endpoints.

Since the playlist dataset is published as an RDF graph on a public Virtuoso instance, accessible and dereferenceable via a persistent URI, and is linked with data from the LOD Cloud, the next step is to explore these additional use-case scenarios which arise from the interlinking, and demonstrate how they can be used in applications developed over the dataset.

4.1 Using Data from the Playlist Dataset

The first question which appears is what kind of information can be retrieved by using only our dataset. It contains consolidated playlist data from different websites, which is enough to enable new use-cases. One such scenario would be finding the songs from a specific artist, along with their titles, their positions in different playlists, and the names of the playlists and radio stations they appear in, at a specific time. In order to get this information, we could use the following SPARQL query:

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX po: <http://purl.org/net/po#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?playlistName ?stationName ?songName
?position
FROM <http://purl.org/net/lmd/data#>
WHERE {
    ?song foaf:name "Arctic Monkeys" ;
    dc:title ?songName ;
    po:featuredInPlaylistEntry ?entry .
    ?entry po:week "31" ;
    po:year "2014" ;
    po:position ?position ;
    po:partOfPlaylist ?playlist .
    ?playlist po:playlistName ?playlistName ;
    po:stationName ?stationName .
}
```

This query finds all the po:Song entities from our dataset which have 'Arctic Monkeys' as an artist name, and retrieves data connected to the instance.

The use of the po:featuredInPlaylistEntry and po:partOfPlaylist properties in this use-case allows for better query performance, compared to the case if we only had the po:playlistEntrySong and po:hasPlaylistEntry properties in the ontology. The partial result of the query executed over our playlist dataset, edited for brevity, is shown in Table 4.

Table 4. Partial results from the SPARQL query.

Playlist	Station	Song	Position
Indie Singles	BBC Radio 1	Do I Wanna Know?	10
Indie Singles	BBC Radio 1	R U Mine?	18
Indie Singles	BBC Radio 1	Why'd you only ...	22

4.2 Using Data from LinkedBrainz and LOD

4.2.1 Using the *po:songInfo* property

The *po:songInfo* property enables us to step out of our playlist dataset and obtain additional data from the LinkedBrainz dataset about the song. For instance, if we want to find out the album (release) for a song from a playlist entry, along with the date it was published, we can use the following SPARQL query:

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX po: <http://purl.org/net/po#>
PREFIX pd: <http://purl.org/net/lmd/data#>

SELECT distinct ?artist str(?songTitle)
str(?releaseTitle) ?releaseDate ?releasePlace
WHERE {
  GRAPH <http://purl.org/net/lmd/data#> {
    pd:JYChLbN-1-3 po:playlistEntrySong ?song .
    ?song po:songInfo ?mbs ;
      foaf:name ?artist .
  }
  SERVICE <http://linkedbrainz.org/sparql> {
    ?mbs dc:title ?songTitle .
    ?record mo:track ?mbs .
    ?release mo:record ?record ;
      dc:title ?releaseTitle .
    ?releaseEvent mo:release ?release ;
      dc:date ?releaseDate ;
      event:place ?place .
    ?place rdfs:label ?releasePlace .
  }
}
ORDER BY ?releaseDate
```

This query starts executing over the local playlist RDF graph, looking for the *po:Song* instance from the selected playlist entry *pd:JYChLbN-1-3*, which represents the occurrence of the ‘Give Life Back to Music’ song by the artist ‘Daft Punk’ in one of the playlists. The detected *po:Song* instance is already linked with a LinkedBrainz song entity, and this entity (its ID) is then sent as a

variable in a subquery for execution at the LinkedBrainz SPARQL endpoint, via SPARQL federation. As a result of the federated call, we obtain the necessary data about the song in question. Since the result set is large, only a part of it is shown in Table 5.

As we see from Table 5, this query can be used by an application for providing a user with more information about the song in question and the album (release) it is part of, by using data not present in our dataset.

Table 5. Partial results from the SPARQL query.

Song	Album	Date	Place
Give Life Back to Music	Random Access Memories	2013-05-17	United States
Give Life Back to Music	Random Access Memories	2013-05-17	Germany
Give Life Back to Music	Random Access Memories	2013-05-17	Netherlands
Give Life Back to Music	Random Access Memories	2013-05-20	United Kingdom

4.2.2 Using the *po:artistInfo* property

Another possible use-case scenario would be to get additional information about the artist of a song featured as an entry in one of the playlists the user is interested in. For instance, a common scenario in an application would be to provide a picture, a description and a website URL for the artist, which can be done with the following SPARQL query:

Arctic Monkeys

Arctic Monkeys are an English indie rock band. Formed in 2002 in High Green, a suburb of Sheffield, the band currently consists of Alex Turner (lead vocals, lead/rhythm guitar), Jamie Cook (rhythm/lead guitar), Nick O'Malley (bass guitar, backing vocals),...

[Read more](#)

Figure 5. Playlist and artist details in the web application.

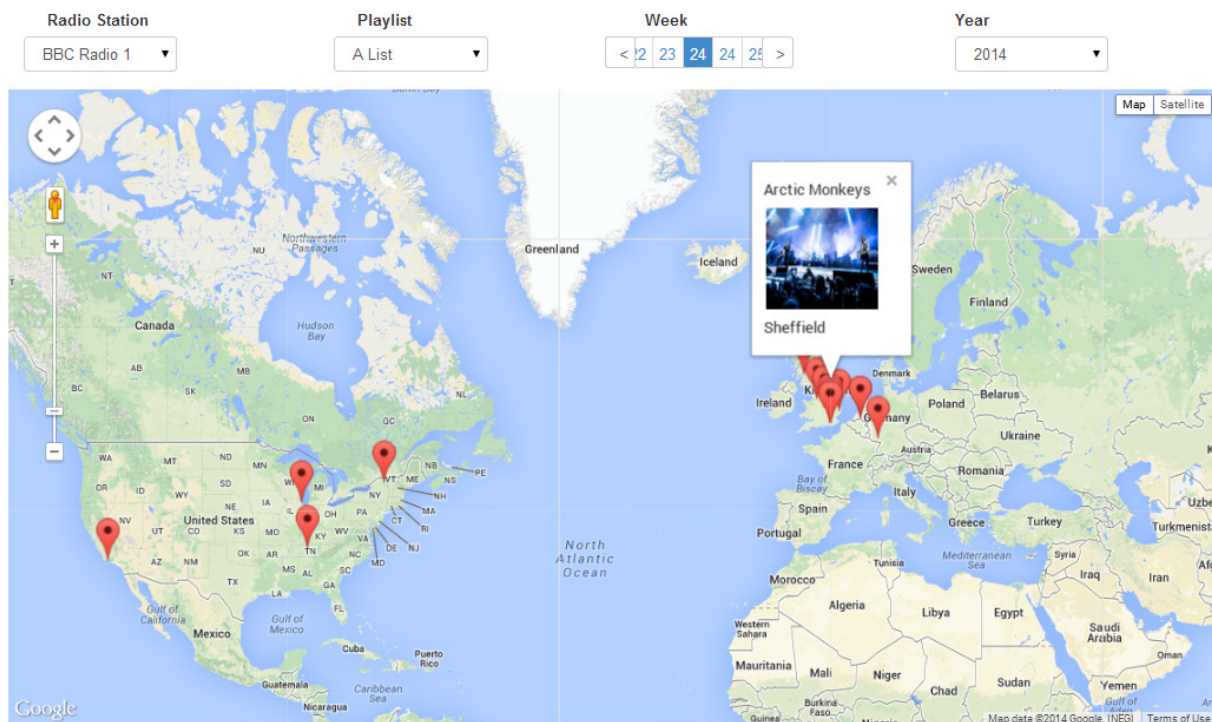


Figure 6. World map view of the artists from a selected playlist / chart, for a specific week and year.

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dbpedia: <http://dbpedia.org/ontology/>
PREFIX po: <http://purl.org/net/po#>
PREFIX pd: <http://purl.org/net/lmd/data#>

SELECT distinct ?thumbnail ?abstract ?website
WHERE {
  GRAPH <http://purl.org/net/lmd/data#> {
    pd:ghQT0qj-1-4 po:playlistEntrySong ?song .
    ?song po:artistInfo ?artist .
  }
  SERVICE <http://linkedbrainz.org/sparql> {
    ?artist owl:sameAs ?dbArtist .
  }
  SERVICE <http://dbpedia.org/sparql> {
    ?dbArtist dbpedia:thumbnail ?thumbnail ;
      foaf:homepage ?website ;
      dbpedia:abstract ?abstract .
    FILTER langMatches(lang(?abstract), "EN")
  }
}

```

This query starts in the local RDF graph, but then continues to retrieve data from the LinkedBrainz and DBpedia datasets, in order to provide the information for the use-case. The result of the example query is shown in Table 6.

The retrieved data is not present in our dataset, but is retrieved from other, distributed data repositories. The data from the result can be used on an artist screen in an application, for example, providing the user with general info about the performer of the song of interest.

Table 6. Results from the SPARQL query.

Thumbnail	Abstract	Website
http://upload.wikimedia.org/wikipedia/commons/thumb/c/c2/Katy_Perry_UNICEF_2012.jpg/200px-Katy_Perry_UNICEF_2012.jpg	"Katheryn Elizabeth Hudson (born October 25, 1984), known by her stage name Katy Perry, is an American recording artist, songwriter, and actress..."	http://www.katyperry.com/

It is important to note that these example queries can be sent as a query string from an application, i.e. the SPARQL endpoint can be used as a REST service. The HTTP GET calls generally have the following format:

```

http://linkeddata.finki.ukim.mk/sparql?query=SPARQLQUERY&format=FORMAT

```

Here, SPARQLQUERY represents the URL encoded SPARQL query, and FORMAT represents the format of the response, such as HTML, XML, JSON, CSV, RDF/XML, N3, Turtle, JSON-LD, etc. The SPARQL endpoint also allows the use of an Accept header for the preferred output format.

Other useful use-case scenarios can be achieved with these interconnected datasets, as well. We could, for instance, collect the social media profile addresses of the artists of interest, find out which label released their most recent album, or make an analytical query and find the artist or label with most songs present on the radio playlists, etc.

These use-case scenarios are meant to be used by developers in various applications from the music and entertainment domain, in order to provide the users with interesting information from the LOD Cloud. These applications have the opportunity to be richer in information than those which use isolated data sources. This will eventually contribute to a better user experience.

5. WEB APPLICATION

In order to demonstrate the feasibility of the use-cases, we developed a web application. It uses our playlist dataset from our Virtuoso instance and aims to provide the end users with basic information about the artists and songs from the available playlists and charts (Figure 5), as well as give them a more analytical insight – a global overview of the countries of origin of the artists present on a given playlist or chart, and allowing an analysis of the weekly dynamics in them (Figure 6).

The web application uses our SPARQL endpoint to query for data from both our dataset and data from the LOD Cloud. One basic use-case is to provide the user additional information about an artist he/she is interested in. This use-case can be achieved by using the list of radio stations and their playlists, and the playlist entries for the current week from the local dataset, along with more artist details – a photo, a short bio, a geo-location of the place of birth/origin of the artist – from the LOD Cloud (Figure 5).

For more analytical users, the web application provides a use-case which offers a global overview of the places the artists from a selected playlist are coming from (Figure 6). By selecting different playlists, the user can gain insight of the differences between radio stations and the varying presence of countries and artists in them. Additionally, by changing the week for one selected playlist, the user can visually witness these dynamics happening from week to week in it. This use-case uses data from the LOD Cloud, as well, in order to get the artist in question, the place of origin or birth, and then its geo-location data.

The scenarios from the web application are in direct support of the idea we initially had: to show that the application of Linked Data principles in the music domain can prove beneficial for the end-users from the domain, by providing more advanced and broader use-cases.

6. CONCLUSION

The concept of Linked Data represents a big advantage in representation and retrieval of structured data from distributed parts of the Web. A large number of communities, companies and other interested stakeholders are taking part in the initiative and are contributing to the expansion of the LOD Cloud [3].

In this paper we described the design of a system which uses an automated workflow to transform music related data from the websites of global radio stations into five-star Linked Data. We developed and published our Playlist Ontology. We also presented and demonstrated novel use-case scenarios, enabled by the interlinked datasets, as a basis for further development of applications and services. As a proof of concept, we developed our own web application which aims to present the benefit of these new use-cases to the end-users.

As we know from [4] and [5], this type of data can help both the business sector and developers, by creating new business value with unique use-cases for applications and services, and the

general public as the end user of those applications and services. Our goal in this paper was to demonstrate that the Linked Data principles offer a bundle of new use-case scenarios in the music domain which were previously either unavailable, or very hard to implement. These use-cases, along with the public dataset itself, can pose a base for further application development by the community and the companies, and can hopefully introduce new business value in the industry.

7. ACKNOWLEDGMENTS

The work in this paper was partially financed by the Faculty of Computer Science and Engineering, at the Ss. Cyril and Methodius University in Skopje, as part of the research project “Semantic Sky 2.0: Enterprise Knowledge Management”.

8. REFERENCES

- [1] C. Bizer, T. Heath, K. Idehen, and T. Berners-Lee, “Linked data on the web,” 17th International conference on World Wide Web, ACM, 2008, pp. 1265-1266.
- [2] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data - the story so far,” International Journal on Semantic Web and Information Systems 5, no. 3, 2009, pp. 1-22.
- [3] T. Heath, and C. Bizer, “Linked Data: Evolving the Web into a Global Data Space,” Synthesis lectures on the Semantic Web: Theory and Technology 1.1, 2011, pp. 1-136.
- [4] T. Berners-Lee, N. Shadbolt, “There’s gold to be mined from all our data”, The Times, 2012.
- [5] V. Kundra, “Digital Fuel of the 21st Century: Innovation through Open Data and the Network Effect”, Joan Shorenstein Center on the Press, Politics and Public Policy, 2012.
- [6] A. Passant, and Y. Raimond, “Combining Social Music and Semantic Web for Music-Related Recommender Systems,” Social Data on the Web Workshop, 2008.
- [7] M. Jovanovik, B. Najdenov, D. Trajanov, “Linked Open Drug Data from the Health Insurance Fund of Macedonia”, 10th Conference for Informatics and Information Technology (CIIT), 2013.
- [8] E. Misheva, B. Najdenov, M. Jovanovik, D. Trajanov, “Open Public Transport Data in Macedonia”, 11th Conference for Informatics and Information Technology (CIIT), 2014.
- [9] B. Najdenov, H. Pejchinovski, K. Cieva, M. Jovanovik, D. Trajanov, “Open Financial Data from the Macedonian Stock Exchange”, ICT Innovations 2014, Advances in Intelligent Systems and Computing, 2014, (in press).
- [10] B. Najdenov, M. Jovanovik, D. Trajanov, “VEO: an Ontology for CO2 Emissions from Vehicles”, ICT Innovations 2014, (in press).
- [11] M. Jovanovik, B. Najdenov, Gj. Strezoski, D. Trajanov, “Linked Open Data for Medical Institutions and Drug Availability Lists in Macedonia”, 3rd International Workshop on Ontologies in Advanced Information Systems, OASIS 2014, Advances in Intelligent Systems and Computing, 2014, (in press).
- [12] Y. Raimond, S. A. Abdallah, M. B. Sandler, and F. Giasson, “The Music Ontology,” ISMIR, 2007, pp. 417-422.