SPARQL-DJ: The MIDI Linked Data Mashup Mixer for Your Next Semantic Party*

Albert Meroño-Peñuela[1], Rick Meerwaldt[1], and Stefan Schlobach[1]

Department of Computer Science, Vrije Universiteit Amsterdam, NL
rickmeerwaldt@hotmail.com, {albert.merono,k.s.schlobach}@vu.nl

Abstract. Many datasets describing musical resources are published today as Linked Data, mainly focusing on metadata, notation, and audio features. However, the availability of musical data as Linked Data is often not enough for *musicians*, who need additional layers of software and queries to accomplish their workflows. Mashups are compositions created by blending two or more pre-recorded songs, and it has been shown that they can be generated by using SPARQL on top of MIDI music represented as Linked Data. In this demonstration, we showcase SPARQL-DJ, a web-based application implementing those layers for the workflow of generating MIDI mashups using Linked Data and SPARQL. The demonstration focuses on three parts: (1) coverage of the mashup composition workflow; (2) scoping of input retrieval for beatmaching by broadening text search and leveraging track annotations; and (3) management of off-beat and dissonance.

Keywords: MIDI, mashups, SPARQL, MIDI Linked Data

1 Introduction

With well-known linked datasets like MusicBrainz, BBC Music and MySpace, *music* is an important player in the Linked Open Data cloud. Most of these datasets represent the *metadata* of music, like bands, musicians, song titles, albums, firms, etc. Some works in the Semantic Web have investigated formal ways of dealing with music itself, like chord representation [2] and the Music Ontology [6]. Deep learning has been applied to link audio with notation [7].

More recently, proposals have been made to represent music *notation* as Linked Data, such as the MIDI Linked Data cloud, a dataset that connects 308,443 MIDI songs by representing their tracks, events, and notes as 10,215,557,355 RDF triples[1] [5]. MIDI (Musical Instrument Digital Interface) is a protocol for

*This demonstration shows and extends the results of the paper "Rick Meerwaldt, Albert Meroño-Peñuela, Stefan Schlobach. *Mixing Music as Linked Data: SPARQL-based MIDI Mashups*. Workshop on Humanities in the Semantic Web - WHiSe II, ISWC 2017".

[1] https://midi-ld.github.io

the interchange of musical information between musical instruments, which can be converted to RDF and back to playable MIDI without data loss [4].

This wealth of information could be a great aid to *musicians* in supporting many of their creative workflows. Concretely, the activity of remixing *existing* music to make new, genuine compositions, also known as *mashups*, is not currently supported by any semantically rich knowledge base. AutoMashUpper, for example, is an "interactive system for creating music mashups by automatically selecting and mixing multiple songs together" based on their harmonic similarity [1]. Mashh! is an online tool where people can find songs and loops and where they can mix their own mashups [8]. The creative process of creating these mashups could be to a great extent automated and scaled up by leveraging semantic representations and Linked Data. In previous work, we have proposed and evaluated a framework for creating mashups using MIDI Linked Data as input and SPARQL queries as method [3]. This demonstration consists of showcasing SPARQL-DJ, a prototype implementing this workflow, at: (1) covering a proposed framework for generating mashups; scoping the search of mashable candidates in two ways: (a) by broadening the search of beatmatching using song metadata, and (b) by leveraging track annotations; and (3) managing off-beat and dissonance in output mashups.

2 Mashups with MIDI Linked Data

Although the creation of mashups involves a more intricate process [3], in this demo we focus on covering the workflow presented in Figure 1. After finding the first pattern, the user searches for its tempo values (step 2). Using the tempo values of the first pattern, the user can search for the second pattern (step 3). After two patterns are found, they are combined in step 4. The result of the query in step 4 can be converted to MIDI, using the MIDI2RDF suite of tools [4] (step 5). The result is evaluated in step 6. Even though at this point the songs are beat-synchronous, it is still possible that they are not in harmony with each other, or that their beat does not match. Consequently, it is possible to search for a different second pattern (step 3). It is also possible to search for the tracks of the songs (step 7) and combine the songs using only certain tracks (step 8). Filtering out certain tracks that cause the dissonant tones can circumvent the problem of the songs not being harmonious. Filtering tracks can also cause for a better mix of instrumentals. After combining the songs using only the desired tracks, the result of the query in step 8 can be converted to MIDI (step 9). The result is evaluated in step 10. If the songs still do not match, different tracks can be combined or filtered out. Contrarily, if the songs do match, a beat-synchronous mashup has been created (step 11).

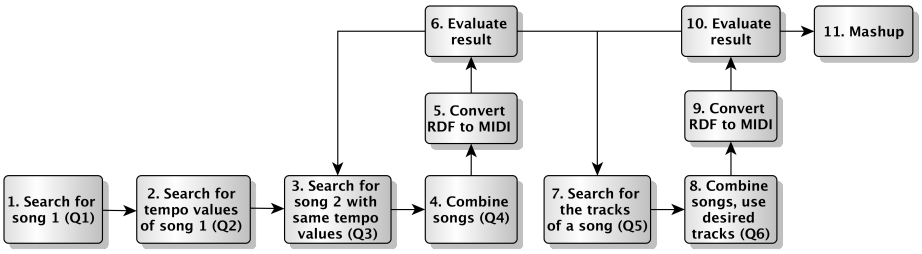


Fig. 1: The mashup creation framework

3 SPARQL-DJ Demonstration

We implement this MIDI mashup creation framework with **SPARQL-DJ** (see Figure 2), a web-based interface for creating MIDI mashups by reusing Linked Data from the MIDI Linked Data cloud [5]. These MIDI data are both *retrieved* and *mixed* into new compositions by exclusively executing SPARQL queries[2] against the MIDI Linked Data cloud SPARQL endpoint[3]. These queries are managed by SPARQL-DJ, and their parameters are supplied from the interface. The resulting MIDI mashups can be played in the browser, or downloaded as either MIDI or Turtle. The demonstration will cover the following use cases.

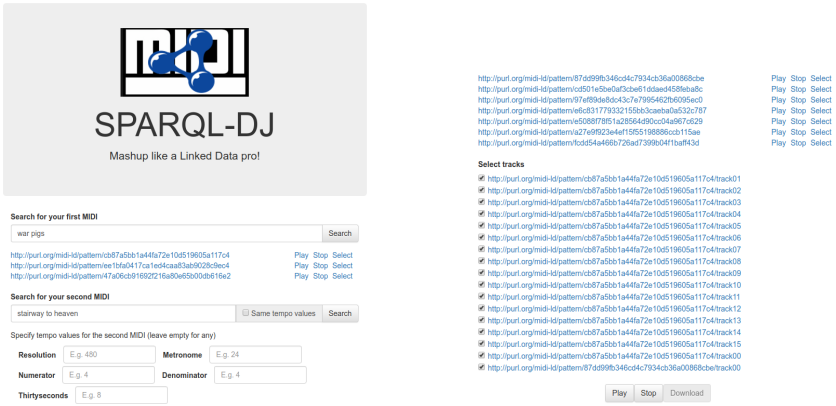
Workflow coverage. Here, the demonstration will cover the steps of Figure 1 by completing the forms in the interface of SPARQL-DJ (Figure 2). First, the user introduces a search string for the first MIDI, getting a number of results that can be inspected by metadata (hovering their URIs), or by playing their MIDI conversions in the browser. After this, users select the first MIDI they wish to mashup. Selecting the second MIDI is analogous, but this time users can search only metric-matching candidates (*same tempo values*), or supply their own metric values. Finally, the user selects which tracks from both MIDIs she wants to have in the mashup. Clicking *play* will create the MIDI mashup and send it to the browser, while clicking *download* will retrieve the MIDI or RDF/-Turtle serialization. This will be demoed with *Michele (Beatles) + Smells Like Teen Spirit (Nirvana)* and *Eine Kleine Nacthmusik (Mozart) + War Pigs (Black Sabbath)*.

Scoping of mashable candidates. We will showcase how the search for beatmatching songs can be broadened in two different ways. First, we will use the additional metadata available in the MIDIs to use the bands' names, their genre, or the collection they come from. Second, we will narrow the number of tracks in the final mashup by filtering them based on their track name annotations.

Managing off-beat and dissonance. In this part, we will show cases in which off-beat and harmonic dissonance might occur, even if the mashable songs are metric compatible; and will discuss about how to manage these.

[2] See queries at `https://github.com/rickmeerwaldt/SPARQLmashup`

[3] `http://virtuoso-midi.amp.ops.labs.vu.nl/sparql`



(a) MIDI lookup, play, and metric filtering. (b) Selection of tracks for the mahsup.

Fig. 2: Screenshot of the SPARQL-DJ interface.

In the future, we will extend SPARQL-DJ's coverage of the mashup creation workflows, we will leverage external links to other datasets for scoping the search of mashable candidates, and we will use harmonic features to manage dissonance.

Acknowledgements. We would like to express our gratitude to Ingrid Arcas for her dedication at testing SPARQL-DJ, her contributed mashups, and her valuable design advice.

References

1. Davies, M.E.P., Hamel, P., Yoshii, K., Goto, M.: AutoMashUpper: An Automatic Multi-Song Mashup System. In: International Conference on Music Information Retrieval (ISMIR). pp. 575–580 (2013)
2. Harte, C., Sandler, M., Abdallah, S., Gómez, E.: Symbolic representation of musical chords: A proposed syntax for text annotations. In: Proceedings of the International Conference on Music Information Retrieval (ISMIR). pp. 66–71 (2005)
3. Meerwaldt, R., Meroño-Peñuela, A., Schlobach, S.: Mixing Music as Linked Data: SPARQL-based MIDI Mashups. In: Workshop on Humanities in the Semantic Web - WHiSe II (ISWC 2017) (2017), under review
4. Meroño-Peñuela, A., Hoekstra, R.: The Song Remains the Same: Lossless Conversion and Streaming of MIDI to RDF and Back. In: 13th Extended Semantic Web Conference (2016)
5. Meroño-Peñuela, A., Hoekstra, R., Gangemi, A., Bloem, P., de Valk, R., Stringer, B., Janssen, B., de Boer, V., Allik, A., Schlobach, S., Page, K.: The MIDI Linked Data Cloud. In: 16th International Semantic Web Conference (ISWC 2017) (2017)
6. Raimond, Y., Abdallah, S., Sandler, M., Giasson, F.: The Music Ontology. In: International Conference on Music Information Retrieval (ISMIR). vol. 422 (2007)
7. Thickstun, J., Harchaoui, Z., Kakade, S.M.: Learning Features of Music from Scratch. In: International Conference on Learning Representations (ICLR) (2017)
8. Tokui, N.: Massh!: a web-based collective music mashup system. In: Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts. pp. 526–527. ACM (2008)