MusicOWL: The Music Score Ontology

Jim Jones
Westfälische Wilhelms-Universität Münster, Germany
Institut für Geoinformatik, Universitäts- und Landesbibliothek
jim.jones@uni-muenster.de

Diego de Siqueira Braga
Westfälische Wilhelms-Universität Münster, Germany
Institut für Wirtschaftsinformatik
diego.siqueira@uni-muenster.de

Kleber Tertuliano
Westfälische Wilhelms-Universität Münster, Germany
Musikhochschule
kleber.tertuliano@uni-muenster.de

Tomi Kauppinen
Department of Computer Science, Aalto University School of Science, Finland
tomi.kauppinen@aalto.fi

ABSTRACT

Managing and analyzing music scores content is a known issue for digital libraries. Despite their inarguable complexity, these contents are often ignored by digital libraries, where music scores are mostly treated as simple digital images. This paper reports on the development and application of MusicOWL, an ontology to formally describe music scores contents for western music. Unlike other efforts on the subject, MusicOWL provides a comprehensive vocabulary for annotating music scores, which covers important music-related information, such as melodies, dynamics, and tonalities. We also report on the Linked Music Score dataset creation, including all necessary steps from scanned music scores to RDF triples, and how this linked music score data enables users to search for music scores by using handy queries, thus translating the needs of many music professionals, data curators and online learners. Additionally, we introduce and analyze the search engine *WWU Music Score Portal*, which uses the approach proposed in this work to improve music scores discovery and support online learning of music.

KEYWORDS

Digital Humanities, Music Scores, Linked Data, OWL.

ACM Reference format:
Jim Jones, Diego de Siqueira Braga, Kleber Tertuliano, and Tomi Kauppinen. 2017. MusicOWL: The Music Score Ontology. In *Proceedings of WI '17, Leipzig, Germany, August 23-26, 2017,* 8 pages.
DOI: 10.1145/3106426.3110325

1 INTRODUCTION

New pieces of music are being constantly composed, and the classics are also often being arranged, which makes the management and use of large repositories of music scores a known nightmare among musicians, musicologists and data curators.

WI '17, Leipzig, Germany
© 2017 ACM. 978-1-4503-4951-2/17/08...$15.00
DOI: 10.1145/3106426.3110325

In most cases, the management of music scores relies entirely on the archivist's knowledge about the music score collection, or the scores are made available via simple text-based searches. Music teachers often have very specific needs when searching for music scores, in order to match their students' level, e.g. amount of chords per measure, pieces written in a specific key or clef, complex dynamics, distribution of particular note lengths, etc. Music experts have even more complex needs, like searching for scores containing a certain theme or rhythmical structure, or even searching for similarities among different music pieces. These detailed and highly complex search criteria make searching for music scores—both from a paper-based repository or when using text-based online searches—a challenging and inefficient experience. Conventional search engines scan for documents often by only using their metadata, which are either related to the document properties (e.g. title, the number of pages or publisher) or to the composer, not to the musical content itself. For instance, the piece *Bachianas Brasileiras No. 5* (see Figure 1), can be found by using typical web search engines by the following search criteria:

Composer: *Heitor Villa-Lobos*
Work: *Bachianas Brasileiras No. 5*
Catalogue of works: *W389.*

However, there are other aspects of music scores, which are important for music experts, that are completely ignored by text-based search engines, for instance: *How many voices / instruments play this piece? How many movements does it have? In which tonality was it written? How many times a particular melody is played along the piece?* or even *What is its level of complexity?* Such data cannot be retrieved by most search engines and are therefore lost inside digital archives. Based on this scenario we present the MusicOWL Ontology, aiming to explore the advantages of the Web Ontology Language (OWL)[1] and Linked Data[2] on efficiently annotating and storing music scores contents. The remainder of this paper proceeds as follows: In Section 2 we discuss on the current state of the art regarding digital music content. In Section 3 we introduce the MusicOWL ontology, describing its main concepts and properties. Section 4 presents the Linked Music Score Dataset, detailing its structure and generation procedure. We also briefly report on the development of the WWU Music Score Portal, a search engine based entirely on the approach presented in this paper. In Section 5

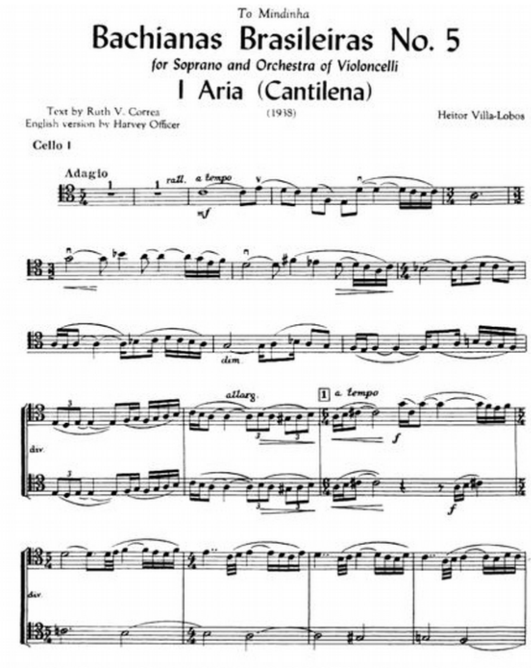


Figure 1: Bachianas Brasileiras No.5 (1st Cello introduction), W389, I Aria (Cantilena), by Heitor Villa-Lobos (1887-1959).

we explore the benefits of our approach through several use cases and query examples, followed by shortcomings and conclusions in Section 6.

2 RELATED WORK

There is a vast amount of projects investigating the benefits of digitally annotating music contents. The Music Ontology[3] compiles a set of classes and properties, and other ontologies, to formally describe acoustic, cultural and editorial information of music content. The Music Ontology provides a wide range of properties for encoding music information, enabling to answer questions like *how a certain piece was recorded* or *how many times and by whom a certain piece was arranged*, but it does not cover detailed information regarding music score contents. MusicXML[4] is a format to encode music scores into XML files, largely adopted among music notation software. Despite the benefits of MusicXML in serializing music scores, it largely lacks semantics and rules for representing music score contents. To date, several research projects addressing music discovery and music recommendation have been conducted [5],[6],[7],[8], but very little effort on music score discovery has been made.

Among the most prominent initiatives on musical sources discovery is the RISM portal[1] (Répertoire International des Sources Musicales). RISM has a large repository of musical sources, enabling users to explore the database using a text-based search or using the advanced feature *On-screen keyboard*. The *On-screen keyboard* enables users to find records by providing the first few notes of the searched music score, the so-called *incipit*. This feature provides a much more efficient search than a simple text-based search, but it is limited only to the beginning of music scores.

The Center for Computer Assisted Research in the Humanities (CCARH),[2] at Stanford University, also offers a portal that allows exploring databases containing musical themes or incipits, the Theme Finder.[3] Besides searching for music scores using notes as search criteria, Theme Finder also enables users to filter results by key, mode and time signature. It has a much more flexible search engine than the previously mentioned approaches, but also lacks support on articulations and dynamics, and is hardly interoperable.

The Music Ngram Viewer[9] has a similar approach to the one presented in this paper. It extracts music contents by using Optical Mark Recognition (OMR) and builds a search engine based on Hadoop and HBase, offering also an intuitive graphical user interface for querying the dataset.[4] Although this approach provides inarguable advances in music score discovery, it does not offer a straightforward way to link its datasets with external data sources, which is an intrinsic advantage of using Linked Data. The Music Ngram Viewer also restricts the maximum length of melodies and bases its query system on a simple sequence of pitches, completely ignoring the duration of the notes. Recent efforts on music scores pattern analyses have been conducted [10], where a tool for analyzing MusicXML files has been proposed, called *Music XML Analyser*.[5] The *Music XML Analyser* offers a platform where users can easily upload MusicXML files to perform quantitative analyses, such as notes, keys, and intervals distributions. The proposed tool also provides a pattern search engine that enables users to look for phrases composed of notes (with their accidentals and octaves) and rhythms, with a maximum length of 12 notes. This approach offers a very intuitive interface for searching for music patterns. However, it ignores other important properties of music scores, such as dynamics and articulations.

3 MUSICOWL

In order to structure music scores data and make them discoverable, in a way that all the previously mentioned issues are addressed, we developed the MusicOWL ontology.[6] This section will describe the main concepts and relationships of the ontology (see Figure 2) and provide some examples of its usage in RDF, together with the correspondent music notation for each proposed concept. The following prefixes are going to be used throughout the paper:

```
xsd: <http://www.w3.org/2001/XMLSchema#>
rdfs: <http://www.w3.org/2000/01/rdf-schema#>
dct: <http://dublincore.org/documents/dcmi-terms/>
mo: <http://purl.org/ontology/mo/>
mso: <http://linkeddata.uni-muenster.de/ontology/musicscore#>
ton: <http://purl.org/ontology/tonality/>
key: <http://purl.org/ontology/tonality/key/>
chord: <http://purl.org/ontology/chord/>
note: <http://purl.org/ontology/chord/note/>
```

mso:Articulation. Articulations define the performance technique, which affects the transition or continuity on a note or between multiple notes. Examples of articulations and their notation in music scores can be found in Figure 3.

[1]https://opac.rism.info/

[2]http://www.ccarh.org/

[3]http://www.themefinder.org/

[4]http://www.peachnote.com/

[5]http://music-xml-analyzer.herokuapp.com/

[6]http://linkeddata.uni-muenster.de/ontology/musicscore/

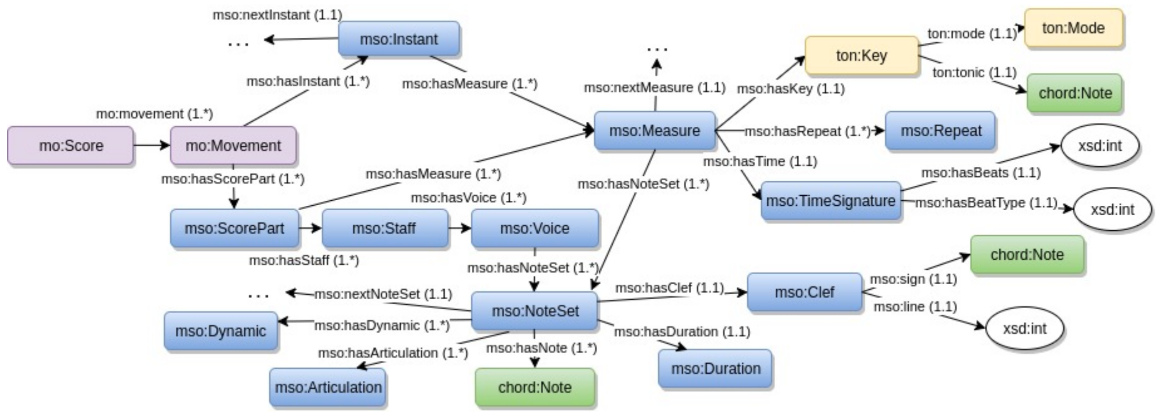


Figure 2: MusicOWL - General Overview

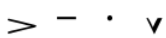


Figure 3: *marcato, tenuto, staccato* and *martellato*.

mso:Clef. A clef is a musical symbol used to indicate the pitch of written notes, by means of placing a note, e.g. the **F** note, on a certain line of the staff. Clefs are divided in four subgroups **mso:GClef**, **mso:CClef**, **mso:FClef** and **mso:NeutralClef**, represented in music notation by the symbols at Figure 4.

Figure 4: G, C, F and Neutral (*Percussion*) Clefs.

mso:Dynamic. Dynamics are instructions in musical notation to musicians about hearing the loudness of a note or phrase. They are divided into two subgroups: **mso:Direction** and **mso:Loudness**. **mso:Direction** dynamics describe the gradual loudness change of notes or phrases, e.g. ***crescendo*** (gradually louder), ***diminuendo*** (gradually softer), see Figure 5. **mso:Loudness** dynamics describe the relative loudness of a note, e.g. ***forte*** (loud), ***fortissimo*** (very loud), ***piano*** (soft), ***pianissimo*** (very soft), see Figure 6.

Figure 5: *crescendo* and *diminuendo* dynamics.

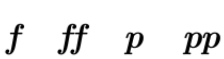


Figure 6: Dynamics: *forte, fortissimo, piano, pianissimo*.

ton:Key. The key determines in which scale or group of pitches a composition or a section of it is written,[11] employing ♯ (sharps) or ♭ (flats) to specific lines of a staff. The number of ♭ or ♯ and the key's graphical representation are defined by the key's position in the circle of fifths (see [4]). In MusicOWL, a *Key* is described based on a certain *Note* (also called tonic) in either *Major* or *Minor* modes. Figure 7 shows the graphical representation of the *Fmajor* and *EMinor* keys. MusicOWL captures the information about *Key* as follows[7]:

```
_:fsharp_major a ton:Key; ton:tonic note:Fs; ton:mode mode:major.
_:eflat_minor a ton:Key; ton:tonic note:Eb; ton:mode mode:minor.
```

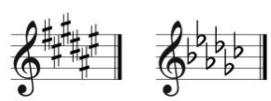


Figure 7: Key Signatures F♯ Major and E♭ minor.

chord:Modifier. A Modifier (also known as *Accidental*) is a symbol that indicates the modification of a pitch, which has been set by the most recent key signature. Most commonly, a pitch can be lowered or raised by a semitone or a whole tone (see Figure 8).

Figure 8: sharp, flat, natural, double sharp and double flat modifiers.

In microtonal music, pitches can also be lowered or raised by a ¼ or ¾ of their natural values, also known as *micro intervals* (see Figure 9).

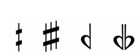


Figure 9: Demisharp, sesquisharp, demiflat and sesquiflat modifiers.

[7]For these we make use of the RDF Turtle format, see http://www.w3.org/TR/turtle/

chord:Note. Represents a pitch of a sound. Notes are composed of a natural value and optionally by an accidental, slightly changing their natural value, e.g. A sharp and E flat notes (see example bellow). Composers often write notes that are supposed to be barely audible or simply not intended to be cleanly played with their full sound. These intentional weak notes are called *Ghost Notes* (also known as *Phantom Notes*). The term *Ghost Note* can be somewhat misleading, since it is rather a property of a note than a note itself. In MusicOWL a *Ghost Note* is therefore classified as an *Articulation.*

```
_:note1 a chord:Note ;
    chord:natural note:A ;
    chord:modifier chord:sharp .

_:note2 a chord:Note ;
    chord:natural note:E ;
    chord:modifier chord:flat .
```

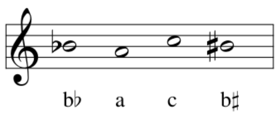


Figure 10: Set of notes.

mso:Duration. Unity that represents the relative duration of a sound. Figure 11 shows a sample of note durations, from a *whole note* until its *128th* fraction.

Figure 11: Note durations.

In medieval mensural notation, the length of a note can be further extended by a double, a quadruple or an octuple of its whole value, see Figure 12.

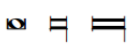


Figure 12: Double, Quadruple and Octuple note durations.

The duration of a note can be prolonged by ½, ¾ or ⅞ of its length by means of placing one, two or three dots right after it, respectively (Figure 13). In MusicOWL, these duration attributes are classified as **mso:Dot**, **mso:DoubleDot** and **mso:TripleDot**.

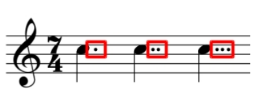


Figure 13: Dotted, double dotted and triple dotted notes

mso:Instant. An Instant is a time unity that contains everything that happens in a music score at a specific moment. It works as a snapshot that captures notes (with their respective durations), articulations and dynamics from measures of all instruments at a distinct moment. Figure 14 shows an example of an instant, containing notes and dynamics from three different staves distributed in two parts.

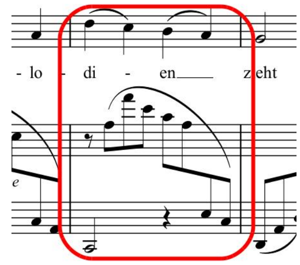


Figure 14: Instant.

mso:NoteSet. *NoteSets* are containers that describe which note (or collection of notes) have to be played and how long they have to played. In music theory, a set of three or more notes played simultaneously is called a chord.[12] The type of a chord can be further defined by the number of notes it has, for instance, triads (three notes), tetrads (four notes), ses Figure 15. In case a *NoteSet* has no *Note* attached to it, it is classified as a *Rest.* Chords can also be further classified depending on their root notes and intervals. Such a classification is supported by the Chord Ontology (see Section 3.1), but goes beyond the scope of this project. A chord can be described as follows:

```
_:noteset a mso:NoteSet; mso:hasNote _:note1, _:note2, _:note3 ;
_:duration a mso:Half .
_:note1 a chord:Note; chord:natural note:G .
_:note2 a chord:Note; chord:natural note:E .
_:note3 a chord:Note; chord:natural note:B .
```

Figure 15: Measure with two NoteSets.

mso:TimeSignature. The time signature determines how many beats are to be contained in each measure, note and the notes' duration:

```
_:timesignature a mso:TimeSignature;
    mso:hasBeats "4"^^xsd:int ;
    mso:hasBeatType "4"^^xsd:int .
```

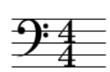


Figure 16: 4⁄4 Time signature.

mso:Measure. Measures (also known as *bars*) are segments of time corresponding to a specific number of beats, in which a particular note value represents each beat, established by its *TimeSignature*.

```
_:measure a mso:Measure;
    mso:hasTime _:timesignature ;
    mso:hasNoteSet _:noteset .
```

Figure 17: Measures (Bars).

mso:ScorePart. Score parts are used to separate the notes of different instruments in a piece or different staves of a single instrument.

```
_:scorepart a mso:ScorePart;
    mso:hasMeasure _:measure;
    mso:hasStaff _:staff .
```

Figure 18: Score part with two staves.

mso:Staff. A set of five horizontal lines and four spaces that represent different musical pitches (contained in voices).

```
_:staff a mso:Staff; mso:hasVoice _:voice .
```

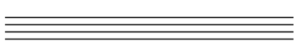


Figure 19: Empty staff.

mso:Voice. In MusicOWL a voice is used to group the sequence of notes of an individual part and staff played by multiple voices. Figure 20 shows an example of a single staff containing two different voices.

```
_:voice a mso:Voice; mso:hasNoteSet _:noteset .
```

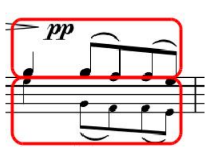


Figure 20: A measure with two voices.

mso:Repeat. Repeats are used to establish musical segments inside a score and are classified as **mso:BeginRepeat**, **mso:EndRepeat**. A **mso:BeginRepeat** indicates that a certain measure is the beginning of a segment inside a piece, which is going to be repeated, and a **mso:EndRepeat** the end of the segment.

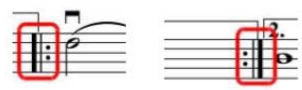


Figure 21: Begin and End Repeat bar lines.

3.1 Used Ontologies

Music Ontology[8]:

provides a vocabulary for publishing and linking a wide range of music-related data on the Web. It has a huge set of classes and properties to annotate information about artists and musical events, such as performances, arrangements.

Chord Ontology[9]: is an ontology for describing chords in musical pieces. It provides a flexible and versatile vocabulary for describing chords and chord sequences in RDF.

Tonality Ontology[10]: provides high-level and low-level descriptors for tonal content in RDF. It offers a complete set of all tonalities that can be found in music scores.

3.2 Ontology Coverage

Music, and consequently also its notation, is a constantly evolving system. Different epochs introduce new expressions, dynamics and playability, and different instruments rely on specific notations in order to fully cover their range of properties. Fully describing such a dynamic system into an ontology is very challenging and is not the intention of the proposed work. The proposed ontology is currently only focusing on the basic and generic elements of a music score presented in this section.

Although playing an important role in music scores, lyrics are also not part of the scope of this work. However, we believe that the usage of existing ontologies,[13] engineered for corpora and linguistic annotation, can be easily used in music score lyrics, enabling analysis also from the linguistics perspective.

4 LINKED MUSIC SCORE DATASET

Like the majority of libraries, the Münster University Library manages historical scores by means of scanning and storing them as digital images.[11] Among other music-related documents, this digital archive has hundreds of historical music scores, with focus on German composers from the early 19th century. This approach facilitates the knowledge preservation of these valuable music scores but entirely relies on text-based search engines to be discovered later on. For instance, the piece *Trois Duos Concertans (1807)*, by Charles Doisy (Figure 22), contains a vast amount of information that can not be found in its metadata, such as melodies, the number of voices, dynamics or articulations.

[8]http://purl.org/ontology/mo/
[9]http://purl.org/ontology/chord/
[10]http://purl.org/ontology/tonality/
[11]https://www.ulb.uni-muenster.de/sammlungen/musik/

Such a digital archive, despite its inarguable historical importance, does not offer a way for musicologists to find music scores by any other means than searching by its limited metadata set,[12] which often only consists of title, composer, length and publisher. Text-based searches also rely on the user's knowledge of the language in which the music score was annotated (French, in the given example), making the efficiency of such search tools rather limited.

Figure 22: Trois Duos Concertans, by Charles Doisy (1807).

To overcome these limitation, we created the initiative Linked Music Score Dataset and the WWU Music Score Portal, which are entirely based on the data structure proposed in this work.

4.1 Dataset Generation

The generation of the Linked Music Score Dataset is divided into the three following steps:

1. Knowledge Extraction: In order to get access the music score's knowledge, we opted to extract it from the scanned music scores using OMR (Optical Music Recognition) and then to convert it to the standard format *MusicXML*.[4] For this task, we opted for the software *PhotoScore & NotateMe Ultimate*,[14] due to its intuitive user interface, but any other OMR software with MusicXML support can be used instead. The PhotoScore OMR engine offers over 99,5%[13] accuracy on most modern PDFs and originals, but when dealing with old scores (sometimes partially damaged or containing old notation systems) this accuracy can be substantially reduced. In either case, the supervision of a specialized librarian is essential to assure data quality in this step.

2. Knowledge Assembling: After converting the scanned music scores to MusicXML, we wrote the converter Music2LOD[14] to parse the MusicXML files and convert them to RDF, using the MusicOWL Ontology. Figure 23 depicts how step 1 and 2 are connected.

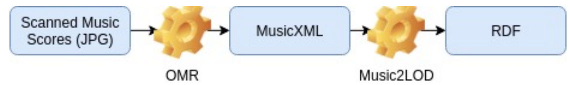


Figure 23: Linked Music Score Dataset Generation.

3. Knowledge Enrichment: The Münster University Library currently stores and manages its scanned documents using the solution Visual Library.[15] This solution provides easy access to documents' images and their meta-data, e.g. Title, Publisher, Author. These meta-data can be accessed in XML format using several standards, specified and maintained by the Library of Congress,[16] e.g. METS, MODS, MARC21. These meta-data can be transformed to RDF using the Dublin Core Terms vocabulary and then be attached to the music score (see Figure 24). The links for the properties *dct:creator*, *dct:publisher*, *dct:contributor* and the resource identifiers are linked to the Linked Open Data services center for North Rhine-Westphalian Libraries[17] (LOBID) and the *Deutsche Nationalbibliothek*[18] (DNB) identifiers, therefore already creating a direct link to these external data sources.

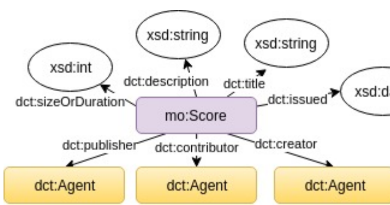


Figure 24: MusicOWL - Score meta-data.

4.2 Dataset Metrics

- **Name:** Linked Music Score Dataset
- **Version date:** 25.06.2017
- **Version number:** 1.0.0
- **Licensing:** Creative Commons Attribution-Non Commercial-ShareAlike 3.0 Unported License.
- **Availability:** As linked data on the Web.[19]
- **Total Scores:** 43
- **Total Statements:** approx. 3.1 million[20]
- **Vocabularies in use:** MusicOWL, RDF, DCMI Terms, Tonality, Chord, Music Ontology.

4.3 WWU Music Score Portal

The WWU Music Score Portal[21] is an initiative from the Münster University to apply the approach proposed in the library's historical music score archive. A significant part of this archive consists of

[12] http://sammlungen.ulb.uni-muenster.de/id/825151
[13] http://www.neuratron.com/photoscore.htm
[14] https://github.com/jimjonesbr/musicowl
[15] https://www.semantics.de/visual_library/
[16] http://www.loc.gov/
[17] http://lobid.org/
[18] German National Library - http://www.dnb.de
[19] http://linkeddata.uni-muenster.de:7200/
[20] http://linkeddata.uni-muenster.de/datasets/
[21] http://linkeddata.uni-muenster.de/musicportal

pieces composed by local composers, and had until now a rather historical than musical relevance to the library users. Many of these composers may be totally unknown to the general public and their pieces may have never been played in the last century. This scenario motivated the Münster University to develop such a tool, so that the users have the possibility to look for melodies and have the opportunity to listen to these rare music pieces. The portal offers an easy to use interface, where users can input melodies (also containing rests) using a virtual piano keyboard - see Section 5.1 for some examples of which queries can be automatically generated by the portal. Users can visualize the typed melodies using several different clefs. The melody search can be easily adapted to simple note sequence search or a rhythmic search by means of clicking on *ignore length* or *ignore pitch*, respectively. After finding a music score containing the given melody, users are also able to play the entire piece directly from the web browser by pressing *play* at the corresponding result set. Although still in a prototypical stage, this portal already shows the potential and acceptance of the approach proposed in this work.

All music engraving components of this portal and the MIDI player, used to play the encoded music scores, were developed using the Verovio Toolkit [15]. Verovio is a flexible and portable framework developed for engraving Music Encoding Initiative[22](MEI) scores into SVG,[16] developed by the Swiss RISM Office[23] and is supported by the Swiss National Science Foundation and the Salzburg Mozarteum Foundation.

5 PROOF OF CONCEPT

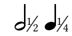
Apart from playing instruments, musicians often face tasks that require a deep understanding of music theory and knowledge of their music score archive. In this section, we briefly discuss some of these needs and demonstrate how they can be easily addressed using the proposed approach, employing SPARQL Queries. For better visualization, every search criteria is also presented with its music notation equivalent.

5.1 Theme Queries

For a reasonable understanding of a music piece, it is often necessary to locate fragments of desired themes, or evaluate if there is a difference in texture (ornaments, articulation, expression and dynamic) or different vocals, e.g. in chamber or orchestral music. In the music research and teaching areas, musicologists often need to harvest their archive looking for complex themes composed only of rhythmical elements, only melodic themes or with rhythm and melody together. To demonstrate how such questions can be answered using the proposed approach, this section shows several examples of search criteria and their correspondent SPARQL Queries.

Query 1 shows an example of how to look for fragments of a piece using only duration elements as input, regardless of note (pitch) or octave:

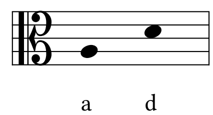
```
SELECT ?score ?measure WHERE {
    ?score mso:hasScorePart ?part .
    ?part mso:hasMeasure ?measure .
    ?measure mso:hasNoteSet ?noteset1 .
    ?noteset1 mso:hasDuration ?duration1.
    ?duration1 a mso:Half .
    ?noteset1 mso:nextNoteSet ?noteset2 .
    ?noteset2 mso:hasDuration ?duration2 .
    ?duration2 a mso:Quarter .}
```



Query 1. Duration search

Query 2 shows an example of how to search for fragments of a piece using only a set of notes as input, regardless of their octave or duration:

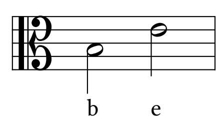
```
SELECT ?score ?measure WHERE {
    ?score mso:hasScorePart ?part .
    ?part mso:hasMeasure ?measure .
    ?measure mso:hasNoteSet ?noteset1 .
    ?noteset1 mso:hasNote ?note1 .
    ?note1 chord:natural note:A .
    ?noteset1 mso:nextNoteSet ?noteset2 .
    ?noteset2 mso:hasNote ?note2.
    ?note2 chord:natural note:D .}
```



Query 2. Note search

Query 3 combines the last two previous query examples, searching for fragments of a music score using notes and their duration. Additionally, the notes' octaves can be also filtered (see Figure 2), in case the searched theme requires an exact pitch:

```
SELECT ?score ?measure WHERE {
    ?score mso:hasScorePart ?part .
    ?part mso:hasMeasure ?measure .
    ?measure mso:hasNoteSet ?noteset1 .
    ?noteset1 mso:hasNote ?note1 .
    ?note1 chord:natural note:B .
    ?noteset1 mso:hasDuration ?duration1 .
    ?duration1 a mso:Half .
    ?noteset1 mso:nextNoteSet ?noteset2 .
    ?noteset2 mso:hasNote ?note2.
    ?note2 chord:natural note:E .
    ?noteset2 mso:hasDuration ?duration2 .
    ?duration2 a mso:Half.}
```



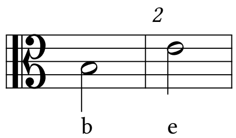
Query 3. Melody and duration search

Query 3 showed an example of how to retrieve the locations of a theme in a score, regardless in how many measures it is encoded. In case a fixed distribution of notes and measures in a phrase is required, one can opt for the searching schema shown bellow on **Query 4**, where the same theme on Query 3 is searched, but with its notes distributed in two different measures:

```
SELECT ?score ?measure1 WHERE {
    ?score mso:hasScorePart ?part .
    ?part mso:hasMeasure ?measure1 .
    ?measure1 mso:hasNoteSet ?noteset1 .
    ?noteset1 mso:hasNote ?note1 .
    ?note1 chord:natural note:B .
    ?noteset1 mso:hasDuration ?duration1 .
```

[22]http://music-encoding.org/
[23]http://rism-ch.org/

```
    ?duration1 a mso:Half .
    ?noteset1 mso:nextNoteSet ?noteset2 .
    ?measure2 mso:hasNoteSet ?noteset2 .
    ?noteset2 mso:hasNote ?note2.
    ?note2 chord:natural note:E .
    ?noteset2 mso:hasDuration ?duration2 .
    ?duration2 a mso:Half .
    ?measure1 mso:nextMeasure ?measure2 .}
```



Query 4. Melody and duration search in fixed measures

5.2 Score Analyses

Gradual increase in music complexity is essential for music learning. Music teachers often need to search through their archives for music scores that match their students' level, which is most of the time done manually. Initiatives like *Sheet Music Plus*[24] contribute to the classification of music score difficulty by establishing guidelines, and manually reviewing the music scores. We argue that these levels of complexity can also be translated into SPARQL Queries, therefore making such a manual evaluation unnecessary.

Query 5 shows an example of such grading, searching for all music pieces in the repository containing 4-note chords:

```
SELECT DISTINCT ?score ?measure (COUNT(?note) AS ?size) WHERE {
    ?score mso:hasScorePart ?part .
    ?part mso:hasMeasure ?measure .
    ?voice a mso:Voice .
    ?voice mso:hasNoteSet ?noteset .
    ?measure mso:hasNoteSet ?noteset .
    ?noteset mso:hasNote ?note .
    ?note chord:natural ?natural .}
GROUP BY ?score ?measure ?voice ?noteset HAVING (?size = 4)
```

Query 6 shows an example of how to retrieve the total amount of measures and to group them per tonality. Such a query is useful for music teachers looking for pieces with a particular pattern of tonality changes. The same principle can be extended to other elements of the music score, e.g. clef or time:

```
SELECT ?tonic ?mode (COUNT(?measure) AS ?total)
WHERE {
?score mso:hasScorePart ?part .
?part mso:hasMeasure ?measure .
    ?measure mso:hasKey ?key .
    ?key ton:mode ?mode .
    ?key ton:tonic ?tonic .
} GROUP BY ?tonic ?mode
```

Query 7 shows an example of how to retrieve the notes distribution of each music score:

```
SELECT ?score ?natural ?modifier (COUNT(?note) AS ?total)
WHERE {
?score mso:hasScorePart ?part .
?part mso:hasMeasure ?measure .
    ?measure mso:hasNoteSet ?noteset .
    ?noteset mso:hasNote ?note.
    ?note chord:natural ?natural .
    OPTIONAL{?note chord:modifier ?modifier}
} GROUP BY ?score ?natural ?modifier
```

[24]http://www.sheetmusicplus.com/help/level-guidelines

6 CONCLUSION AND FUTURE WORK

In this paper, we explore the benefits of using Linked Data technologies to semantically annotate and improve discovery of music scores. We proposed a procedure to convert existing music scores, either in XML files or scanned images, into RDF using MusicOWL. This method was implemented in music scores managed at the Münster University Library's archive (historical and modern music scores), resulting in the Linked Music Score Dataset. Further on, we presented several example queries to evaluate the level of expressiveness that our approach supports. The results indicate that applying Linked Data technologies to music scores discovery is entirely feasible and very promising, enabling users to find music scores providing search criteria in a level of detail not much explored before. We also demonstrated that our approach enables users to find music scores based properties like, the number of chords, specific dynamics or key changes, which opens the possibility to find and classify music scores regarding their level of complexity. Future work should focus on extending the ontology to support sophisticated musical analysis, e.g. melodic and harmonic analysis. Finally, we intend to explore the possibility of adding RDF export to conventional music notation software, such as MuseScore.[17] It would enable composers or musicologists to easily generate the meta-data necessary to our approach.

ACKNOWLEDGEMENTS

This work has been funded by the Münster University Library (*Universitäts- und Landesbibliothek Münster*).

REFERENCES

[1] S. Bechhofer, "Owl: Web ontology language," in *Encyclopedia of Database Systems*. Springer, 2009, pp. 2008–2009.
[2] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data-the story so far," *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pp. 205–227, 2009.
[3] Y. Raimond, S. A. Abdallah, M. B. Sandler, and F. Giasson, "The music ontology." in *ISMIR*. Citeseer, 2007, pp. 417–422.
[4] M. Good, "Musicxml for notation and analysis," *The virtual score: representation, retrieval, restoration*, vol. 12, pp. 113–124, 2001.
[5] P. Knees, T. Pohle, M. Schedl, and G. Widmer, "A music search engine built upon audio-based and web-based similarity measures," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 447–454.
[6] R. Iwamura, "Music search by melody input," Feb. 13 2001, uS Patent 6,188,010.
[7] M. Woo, "Music search engine," Jan. 13 2004, uS Patent 6,678,680.
[8] C. S. Mesnage, A. Rafiq, S. Dixon, and R. P. Brixtel, "Music discovery with social networks," in *Workshop on Music Recommendation and Discovery*, 2011, pp. 1–6.
[9] V. Viro, "Peachnote: Music score search and analysis platform." in *ISMIR*, 2011, pp. 359–362.
[10] M. Burghardt, L. Lamm, D. Lechler, M. Schneider, T. Semmelmann, and C. Wolff, "Tool-based identification of melodic patterns in musicxml documents," 2016.
[11] J. Clough and G. Myerson, "Musical scales and the generalized circle of fifths," *The American Mathematical Monthly*, vol. 93, no. 9, pp. 695–701, 1986.
[12] Benward and Saker, *Music: In Theory and Practice*. ISBN 978-0-07-294262-0, 2003.
[13] C. Chiarcos and M. Sukhareva, "Olia–ontologies of linguistic annotation," *Semantic Web*, vol. 6, no. 4, pp. 379–386, 2015.
[14] B. Finn and J. Finn, "Sibelius: The music notation software," 2001.
[15] L. Pugin, R. Zitellini, and P. Roland, "Verovio: A library for engraving mei music notation into svg." in *ISMIR*, 2014, pp. 107–112.
[16] P. Roland, "The music encoding initiative (mei)," in *Proceedings of the First International Conference on Musical Applications Using*, vol. 1060, 2002, pp. 55–59.
[17] T. Bonte, "Musescore: Open source music notation and composition software." Technical report, Free and Open source Software Developers' European Meeting, 2009. http://www. slideshare. net/thomasbonte/musescore-at-fosdem-2009, 2009.