Revue des Traditions Musicales

ISSN : 3071-3226
Vol. 18, No. 1 | 10.51300/RTM-2024-140

LUMINOUS INSIGHTS

Article de recherche

Automation of the Musical Generative Grammar Encoding Process for Modal Monodies

Automatisation du processus d'encodage de la grammaire générative musicale pour les monodies modales

Charbel El Achkar and Talar Atéchian

Ticket Lab, Faculty of Engineering and Technology, Antonine University, Baabda, Lebanon

RÉSUMÉ

Computer-assisted analysis of musical notation can focus on surface structure, or it can probe the underlying structuring of the music being analyzed, thereby revealing the generative musical grammar of that music. This article proposes an automated procedure for generative musical grammar rewriting of digitally encoded modal monodies, based on modal semiotic theory. This procedure takes the form of an ontology that structures the knowledge extraction process—particularly semantic knowledge—of a pattern analysis algorithm for modal music traditions from Western Asia. In addition to the mandatory elements describing music scores, the proposed ontology relies on contextual elements and attributes for pattern analysis. The ontology thus supports semantic information retrieval and content analysis of music scores. The entire mechanism is illustrated by explaining the workflow of the ontology embedded in a music encoding platform for modal music. Furthermore, a music score converter called MEI2JSON is presented, which converts music scores encoded in the MEI format to JSON format. This converter shares the same support for modal music scores as the proposed ontology. The conversion process is demonstrated by evaluating performance analysis, data quality, and storage of the proposed converter against a combined approach consisting of two state-of-the-art converters.

ABSTRACT

L'analyse des notations musicales assistée par ordinateur peut s'intéresser à la structure de surface, comme elle peut sonder la structuration sous-jacente de la musique analysée, donc faire état de la grammaire générative musicale de cette musique. Cet article propose une procédure automatisée de réécriture grammaticale générative musicale de monodies modales

MOTS-CLÉS

modal semiotics, score converter, music encoding, musical grammar rewriting, knowledge extraction, generative musical grammar

KEYWORDS

sémiotique modale, convertisseur de partitions, encodage musical, réécriture grammaticale musicale, extraction de connaissances, grammaire générative musicale

ARTICLE HISTORY
Published : 24 May 2026

Corresponding author :
Talar Atéchian | talar.atechian@ua.edu.lb | Ticket Lab, Faculty of Engineering and Technology, Antonine University, Baabda, Lebanon

encodées numériquement, qui est basée sur la théorie sémiotique modale. Il s'agit d'une ontologie qui structure le processus d'extraction de connaissances notamment sémantiques d'un algorithme d'analyse de motifs musicaux modaux relevant de traditions musicales d'Asie occidentale. En plus des éléments obligatoires qui décrivent les partitions musicales, l'ontologie proposée s'appuie sur des éléments contextuels et des attributs pour l'analyse des modèles. L'ontologie prend alors en charge les processus de recherche d'informations sémantiques et d'analyse du contenu des partitions musicales. L'ensemble du mécanisme est illustré en expliquant le flux de travail de l'ontologie intégrée dans une plate-forme d'encodage musical pour la musique modale. De plus, est présentée une proposition de convertisseur de partitions musicales nommé MEI2JSON pour convertir les partitions musicales encodées au format MEI au format JSON. Le convertisseur partage le même support pour les partitions de musique modale que l'ontologie proposée. Le processus de conversion est illustré en évaluant l'analyse des performances, la qualité des données et le stockage du convertisseur proposé par rapport à une approche combinée composée de deux convertisseurs à la pointe de la technologie.

1. Introduction

Many music scores are encoded for analysis purposes using XML formats such as MusicXML (Good, 2001, p. 113–124) or MEI (Roland, 2002, p. 55-59). The analysis of a music score consists of extracting its underlying features. Therefore, many ontologies have been proposed to structure music score content for better information retrieval (Cherfi et al., 2017 ; Jones et al., 2017). Researchers have developed text-based platforms to archive and restore music scores. These platforms store music scores based on their metadata. Thus, users can search for scores using criteria such as the composer's name, publication date, title, etc.

However, there are other important features for music analysts—known as semantic features—that are not covered by many platforms. These include the number of instruments in a music score, the tonality in which it is written, as well as the number and order of occurrence of specific notes within a music score. All these features, and many more, hold essential information that, when organized, constitute criteria for comparing and retrieving music scores.

In this article, we present a new ontology named MusicPatternOWL that aims to leverage the advantages of the Web Ontology Language (OWL) for encoding and annotating modal (West Asian) music scores based on their pattern analysis, in relation to the generative musical grammar underlying modal monodies. The ontology is built upon the schema structure of the MEI format.

Moreover, we present a new data converter named MEI2JSON that aims to convert music scores encoded in MEI to JSON format while preserving their modal (West Asian) music score content. The converter is based on the proposed MusicPatternOWL ontology, in addition to a modified schema of MEI capable of providing structured knowledge extraction of music score elements and attributes for modal (West Asian) music encoded in MEI. MEI2JSON is also capable of performing MEI-to-JSON conversion without requiring the combination of multiple converters from different sources.

The remainder of this paper is organized as follows : Section 2 discusses recent music-related ontologies and converters. Section 3 introduces the proposed MusicPatternOWL, presenting its structural aspects and score analyses through a preliminary proof of concept. Section 4 explores the implementation of the MEI2JSON converter and compares its performance against a combination of two existing converters through experiments, followed by a conclusion in Section 5.

2. Related Work

The digitization of music analysis theories has proven to be a collaborative opportunity between developers and music analysts. Developers provide encoding solutions, while musicologists can obtain error-free analytic results in a reduced amount of time. A music theory named "Modal Semiotics" is proposed by Nidaa Abou Mrad (2016) for analysing traditional modal monodies (T.M.M.) of West Asian and Mediterranean cultures, from the perspective of generative musical grammar. In Asmar et al. (2018, p. 95-103), a semantic-based platform is proposed to encode and analyse T.M.M. The encoder analyses T.M.M. music scores by extracting their underlying semantic features and, in the end, adds a custom module to the MEI schema to provide an explicit visualization of the encoding. It is important to mention that the extraction process consists of applying two input matrices that correspond to several combinations of music score patterns. Thus, it is significant to develop an ontology to structure the elements of a music score.

Ontologies play a key role in establishing semantic relationships between analytical theories in music. They enable automated reasoning over data and offer more coherent navigation, facilitating transitions from one theory to another. Recent research has explored the development of ontologies for representing music score content. For instance, Jones et al. (2017) propose an ontology that formally describes music score content within the context of Western tonal music, leveraging the Web Ontology Language (OWL) for annotation and storage. Similarly, Cherfi et al. (2017) introduce an ontology aimed at integrating semantic music elements to support knowledge extraction and management. Their approach provides a generic framework for representing, extracting, linking, and searching music theories.

Many music scores are encoded using symbolic formats such as MEI (Roland, 2002, p. 55–59) and MusicXML (Good, 2001, p. 113–124). These formats—MEI in particular—are XML-based and rely on XML schemas to define the structure of their elements and attributes. In this context, frameworks like JXML2OWL (Rodrigues et al., 2008, p. 808–819) were introduced to manually map XML schemas to existing OWL ontologies, and later to automate the transformation of XML instances into individuals of the mapped ontology. Such frameworks enabled the efficient conversion of syntactic data representations (using XML) into semantic ones (using OWL). This transformation supported inference over knowledge-based models, thereby improving data exchange and integrity.

Another mapping solution, proposed by Lacoste et al. (2011, p. 145–154), involved an efficient framework for automatically generating ontologies directly from XML instances. This approach provided a robust description of both the OWL model and the XML instance files. The emergence of both manual and automatic mapping frameworks between OWL ontologies and XML schemas made XML-encoded data accessible to Semantic Web applications already connected to OWL ontologies. This development paved the way for tools such as SPARQL2XQuery, which highlight the adjacency and interoperability of OWL and XML. Consequently, the framework introduced by Bikakis et al. (2009) enabled the evaluation of SPARQL queries over XML data following the mapping of XML to OWL schemas.

Mapping frameworks proved particularly valuable when the transformation rules between XML and OWL schemas could be saved and reused as needed by storing them in XSL stylesheets. The use of XSL files as containers for mapping rules encouraged their adoption across multiple data format converters, as they ensured data conversion without loss of quality through direct schema-to-schema mapping based on datatypes and property rules.

In the context of music-related research, a toolkit called music21 was introduced by Cuthbert and Ariza (2010, p. 637–642). Although it did not use XSL stylesheets, it provided software tools for both musicians with limited programming experience and programmers to analyse, search, and transform symbolic music scores. The toolkit supports the conversion of several symbolic formats, including MEI, MusicXML, and MIDI (The MIDI Manufacturers Association, 1995).

With the evolution of the MEI format, Verovio—a music engraving library—was developed by Pugin et al. (2014, p. 107–112) to produce visual representations of MEI-encoded music scores in SVG. This library also enabled bidirectional conversion between MusicXML and MEI, relying on the MEI XSL stylesheets available in the MEI encoding tools on GitHub.

The conversion capabilities of Verovio were limited, as it focused on the main elements and attributes of music scores while excluding others. Another MEI-related conversion framework, named Meico, was introduced by Berndt et al. (2018) to provide a novel tool for processing MEI-encoded music scores. Meico facilitated the conversion of MEI data into multiple symbolic formats, including MusicXML. However, this conversion relied on the same XSL stylesheets used in Verovio and consequently also failed to convert all the elements and attributes of a music score encoded in MEI.

A further study by Alvaro and Barros (2010) focused on developing a music composition system called Computer Music Cloud (CMC) along with a dedicated data representation format known as MusicJSON, designed to efficiently compose and store music scores within the computer music cloud. MusicJSON served as a music interchange tool between different services of the CMC and was also used as a music data unification tool by converting several symbolic formats—including MusicXML—into a JSON-based music representation format.

3. The MusicPatternOWL ontology

3.1. *Structural Aspects*

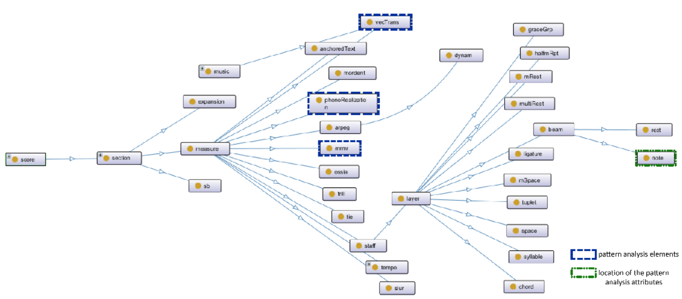

Figure 1 : MusicPatternOWL – General Overview

Based on the MEI schema, the MusicPatternOWL ontology aligns with the contribution proposed by Asmar et al. (2018, p. 95–103). It introduces new elements and attributes to provide a structured framework for analysing note patterns within a music score (see Figure 1). Before detailing the elements and attributes related to pattern analysis, it is important to note that a single music score contains multiple measures, each of which is composed of several notes.

snr (Syllabic Nuclear Reduction or Musical Phonologic Nuclear Reduction) – Attribute of the note element. It takes only two values : α or β. The pattern analysis algorithm developed by Asmar et al. (2018, p. 95–103) considers the final note of a music score as the main input element. Subsequently, each note in the score is assigned a value of either α or β. According to the phonological component of the theory proposed by Abou Mrad (2016), The value α indicates that the encoded note's pitch belongs to the subset of odd-ranked degrees, whereas β indicates that the encoded note's pitch belongs to the subset of even-ranked degrees. Thus, at the level of a music score containing multiple measures of notes, a pattern composed of α and β values is derived. This constitutes a melodic phonological rewriting of the analysed score.

1. **mnr** (Metasyllabic Nuclear Reduction or Musical Morphological Nuclear Reduction) – Attribute of the formula element. It accepts only boolean values and represents a rhythmic parameter corresponding to morphological rewriting. A value of true is assigned to a note that both possesses an snr attribute and has the longest duration among its adjacent notes. A value of false is assigned if either of these conditions is not met. The longest duration is determined using a note matrix provided by the music analysts. Consequently, applying different matrices to the same music score yields distinct pattern analysis possibilities.

2. **mrmr** (Morphophonological Rhythmic and Melodic Rewriting) – Child of the measure element. It contains matrices and mathematical equations. This element stores the various patterns extracted from the music score, based on the melodic attribute (snr) and the rhythmic attribute (mnr) described above.

3. **phonoRealisation** (Phonological Realisation) – Child of the measure element. Like the mrmr element, this element contains matrices and equations, but it encodes exclusively the underlying phonological features of a music score.

4. **vecTrans** (Vector Transcoding) – Child of both the measure element and the melodic element. This element contains vectors generated by combining the mnr attributes. The result is a vector at the mea-

sure level and a series of vectors at the level of the entire music score (within the music element). The generation of these vectors constitutes the final step of the algorithm proposed by Asmar et al. (2018, p. 95–103), providing meaningful knowledge extraction for musicians.

In addition, the **number** attribute is included for the **mrmr**, **phonoRealization**, and **vecTrans** elements. It serves as an identifier for each measure within a music score.

The remaining elements of MusicPatternOWL shown in Figure 1 correspond to typical components of a music score. Based on the MEI schema developed by Roland (2002), our ontology achieves full coverage of any element or attribute required to digitally encode or annotate music scores in the MEI format. It should be noted that the elements and attributes discussed in this section are optional, as we intend to extend this ontology to also support pattern analysis in Western music.

3.2. *Score Analyses*

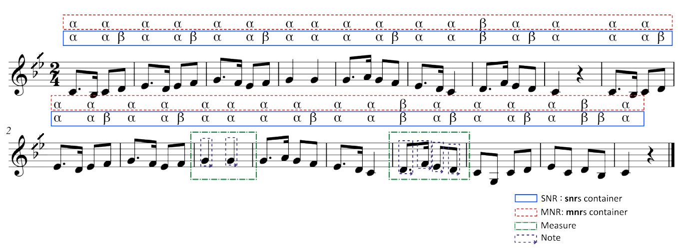

Figure 2 : SVG output from the MM analyser

The music encoding algorithm proposed by Asmar et al. (2018, p. 95–103) served as the starting point for creating MusicPatternOWL. In brief, the algorithm has been integrated into a music encoding platform for the traditional modal monodies of the *Mašriq* called the "MM Analyser". The platform expects as input an MEI document along with the two matrices mentioned earlier in Section 2. It outputs another MEI document, along with a PDF file containing all the corresponding analysis, generated using Verovio (Pugin et al., 2014, p. 107–112) and SVG processing to place alphas and betas above the notes.

Analysing music scores reveals underlying features by assigning α and β values to the snr attribute. The snr values of a music score are grouped in SNR, while mnr values set to "true" receive the same snr value as their corresponding note and are grouped in MNR. This process generates a pattern of α and β values—first grouped in SNR, then in MNR—for each music score (see Figure 2) (Abou Mrad, 2016, ch. 7). Consequently, the proposed ontology tracks the analysis at every progressive encoding step, providing structured knowledge extraction. It covers the entire workflow, from pattern establishment to vector generation, through the elements and attributes already presented. Additionally, it prevents any false or abnormal insertion of musical notation via its elements, managing not only the analysis itself but also the structure and constraints of the music score. It is important to note that the entire analysis process is expressed through mathematical expressions. This underscores the need for a rule-based ontology to validate all relevant properties and ensure error-free platform behavior.

4. The MEI2JSON Converter

4.1. *Motivation*

The MM analyzer presented in Asmar et al. (2018, p. 95-103) and the MusicPatternOWL ontology introduced in El Achkar & Atéchian (2020, p. 17-20) addressed one of the most fundamental issues in music-related platforms : the lack of support for modal music encoding and analysis. The MM analyzer facilitated the encoding and analysis of modal music scores, while MusicPatternOWL supported this analytical process by ensuring error-free knowledge extraction at each progressive step of the encoding. At this stage, we were able to export lossless music scores encoded in the MEI format.

The advantages of combining AI techniques with music-related platforms motivated us to integrate these techniques and improve the MM analyzer. As with any AI use case, data must be prepared and simplified as much as possible before being ingested for training in neural networks. Therefore, it was necessary to convert our MEI exports to another data format, since MEI contains many elements and attributes that can be reduced depending on the use case. Based on the music-related converters reviewed in the related work, we observed the absence of a converter capable of transforming MEI music scores into JSON format. Furthermore, a critical requirement remained unaddressed : converting music scores without compromising data quality or introducing errors.

All the reasons mentioned above motivated us to create the MEI2JSON converter, capable of transforming MEI music scores into a simplified JSON format while preserving data quality and reducing data manipulation errors, particularly for Modal music score datasets.

4.2. MEI2JSON Process

Figure 3 presents the MEI2JSON converter through an activity diagram. The first part of the diagram illustrates the progressive steps of the MEI2XML component required to achieve a successful conversion (MEI to XML). The converter retrieves an MEI score from an MEI file (with a .mei extension), loads the custom XSL stylesheet created for this purpose, and converts the MEI file to XML if possible. The custom XSL stylesheet, named 'mei2xml.xsl', is responsible for handling the required conversion. This stylesheet must be loaded by an XSLT processor to perform the conversion. We therefore use the Saxon XSLT and XQuery processor (Kay, 2010), based on its prior use in most of the converters reviewed in the related work.

In the event of a successful conversion, the generated XML file is passed to the second component for further processing. Otherwise, the system logs the errors, allowing us to easily identify and resolve the issues that caused the conversion failure. The generated XML file then proceeds to the XML2RDF component. As with the previous component, the 'xml2rdf.xsl' stylesheet proposed in Breitling (2009) is loaded using the Saxon processor to apply the corresponding conversion to the XML file. Similarly, the generated RDF file moves to the next component upon success, and in case of failure, the error logs guide the user in resolving the encountered problems.

Finally, the generated RDF file is loaded into the RDF2JSON component using the RDFLib library. This library provides powerful parsers and serializers to load the knowledge graph from RDF/XML data. Once loaded, the RDF file can be queried through custom SPARQL queries to extract the semantic information needed for preprocessing purposes. The SPARQL query can then be customized according to the use case. In the event of a successful query, the results are sorted and formatted as a JSON file for output. The RDF2JSON component ensures the validity of the JSON file by applying schema syntax definitions, such as the JSON schema proposed in Pezoa et al. (2016). Thus, the MEI2JSON converter progresses through several steps, moving from one component to another, to achieve a successful conversion of MEI scores to JSON.

Note that the MEI2JSON converter is currently implemented using the Python language (Van Rossum et al., 2009) ; however, it can be implemented in other programming languages as well, since we load the XSL stylesheet via command-line usage of the Saxon library. Furthermore, RDF-related libraries are available in many programming languages, which helps provide enhanced coverage for the MEI2JSON converter.

4.3. Experiments

The performance of converters is typically evaluated by calculating their complexity and ensuring that they preserve the quality of the data produced from the transformation. In Figure 4, we present four different data quality metrics applied to both the Meico+MusicJSON converter and the MEI2JSON converter to assess their ability to preserve data quality on the dataset used. The metrics for MEI2JSON are visualized using orange bars in the histogram, while those for Meico+MusicJSON are shown using blue bars.

Before explaining the quality metrics, we note the use of the Jsonix mapping library to obtain a JSON schema from the modified MEI schema proposed by Asmar et al. (2018). We then used the resulting JSON schema and the MusicJSON schema to evaluate the outputs of the Meico+MusicJSON and MEI2JSON converters, respectively, against each metric. It is worth mentioning that the mandatory elements in this experiment concern the "note" element and its attributes, including those specific to modal music scores. Below, we present the assessment metrics used to compare both approaches.

Availability is a metric used to measure whether all the necessary elements of a music score are present in a specific dataset. The dataset in question concerns the output generated by both converters. We measured the availability of both approaches by calculating the percentage of music score fields that have values entered into them. MEI2JSON resulted in an availability percentage of 98.2%, while Meico+MusicJSON yielded 63.9%. The gap between both results is mainly due to the lack of support for the modified MEI schema in the Meico+MusicJSON approach.

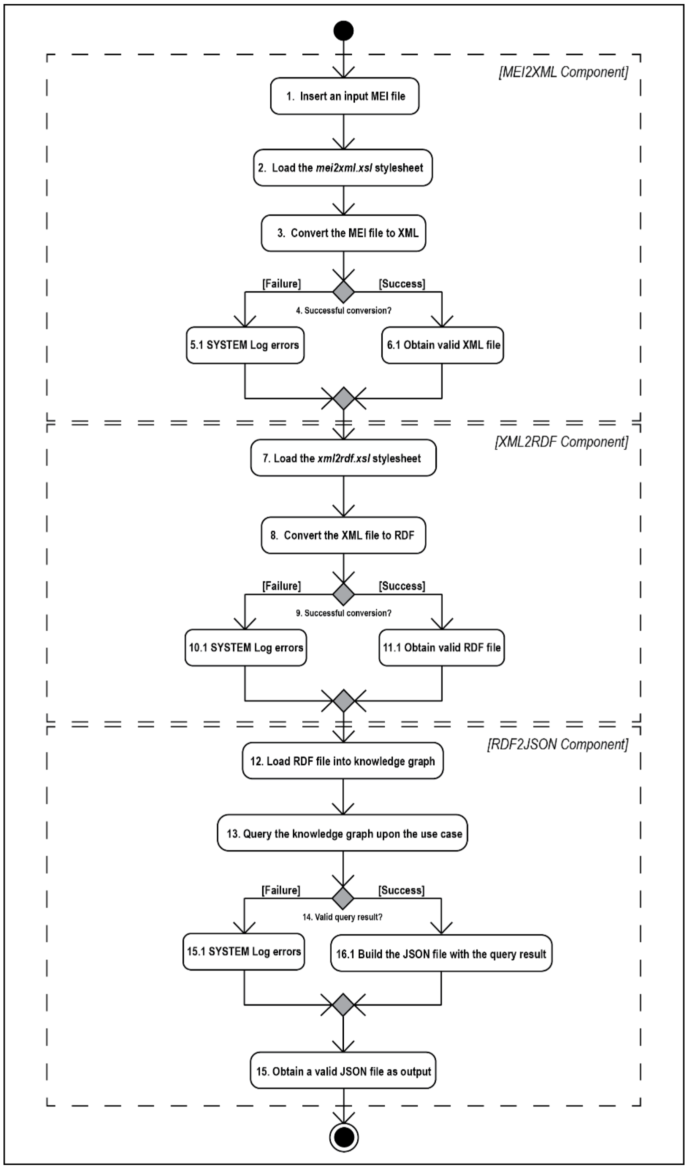

Figure 3 : MEI2JSON Activity Diagram

This lack of support leads to discarding undefined elements and preserving only standard tonal music score elements. In this case, the undefined elements are those related to modal music scores.

Accuracy is a metric used to evaluate the correctness of the music score in question. We measured the accuracy of both approaches by calculating the percentage of correctly converted music score elements compared to the initial values. Since the future use of the converted music

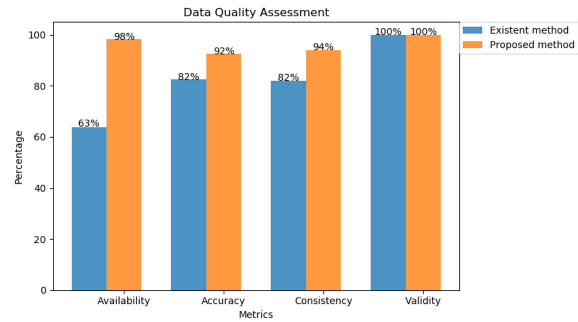

Figure 4 : Histogram – Data Quality Metrics

scores lies in the AI field, we chose to estimate accuracy using the *accuracy_score* function provided by Pedregosa et al. (2011). Considering the use of several essential elements for encoding music scores, we estimated accuracy as a multilabel approach. We calculated the accuracy at the level of multiple elements, each element containing multiple labels. As shown in the equation below, accuracy is the sum of accuracies at the level of each music element divided by \(N \) — the number of music score elements used. The *true_label* refers to the labels of the initial music elements before conversion, and the 'output_label' refers to the labels of elements after JSON or MusicJSON conversion. It is important to mention that since music score elements have labels encoded as characters in MEI, we had to use the *LabelEncoder* function provided by Pedregosa et al. (2011) to convert characters to numeric values. This way, the element labels become compatible for use in the *accuracy_-score* function.

For example, before using the *accuracy_score* function, we use the *LabelEncoder* to transform the labels of the *PitchName* element from [a, b, c, d, e, f, g] to [0, 1, 2, 3, 4, 5, 6]. This allows us to first calculate the accuracy of *PitchName* using *accuracy_score*, then compute the accuracy of the remaining elements, sum all the accuracies, and divide by the number of elements used. Finally, we multiply the resulting accuracy by 100 to obtain the accuracy value as a percentage.

$$Consistency = \frac{\sum v\_measure\_score(true\_label, output\_label, \beta)}{N} \times 100 \tag{1}$$

The Meico+MusicJSON resulted in an accuracy of 82.4%, and the MEI2JSON an accuracy of 92.5%.

Consistency is a metric used to evaluate the synchronicity of a music score in terms of data types and schema structure. We measure this metric by calculating the percentage of data types that match across different records. We use the schema structure of both approaches to detect changes in structure and data types when passing from one component/converter to another.

Similar to the accuracy calculation, we took the same approach to calculate consistency at the level of data types per music score element. Therefore, we used the *LabelEncoder* function to transform data type values into numeric labels and employed the *v_measure_score* function provided by Pedregosa et al. (2011) to estimate the consistency of music score elements.

By definition, the *v_measure_score* clusters labels given a ground truth. In our case, it clusters the data type labels of each element, reflecting the consistency of those elements.

The *v_measure_score* function takes the following parameters : *true_label*, which represents the element's data type labels before using the MEI to JSON or MusicJSON converters ; *output_label*, which represents the element's data type labels after MEI to JSON or MusicJSON conversion ; and β, the ratio of weight attributed to homogeneity and completeness. We leave this value at its default, meaning that the resulting score gives equal weight to both homogeneity and completeness. The v_measure_score returns a score between 0.0 and 1.0. The higher the score, the better the consistency.

Once the *v_measure_score* is calculated for all essential elements of a music score, we sum all the resulting scores, divide the total by \(N \) (the number of music score elements used), and finally multiply the result by 100 to obtain the final consistency value as a percentage.

$$Accuracy = \frac{\sum accuracy\_score(true\_label, output\_label)}{N} \times 100 \quad (2)$$

As shown in Figure 4, the consistency percentage of Meico+MusicJSON is 82%, while that of MEI2JSON is 94%. This slight improvement of MEI2JSON over the combined approach is due to the presence of the MusicPatternOWL ontology in the first and last converter, which helps structure and filter the music score elements in question.

Validity is a metric used to measure how well data conforms to the required value attributes. We measured validity by calculating the percentage of music score elements and attributes that have values within the domain of acceptable values. We used the JSON schema of both approaches, along with the syntax definition proposed by Pezoa et al. (2016), to calculate the validity metric. Both approaches resulted in a validity percentage of 100%. This high percentage is due to MusicJSON's built-in validator in the first approach and the presence of the MusicPatternOWL ontology in the second.

5. Conclusion

In this article, we presented the MusicPatternOWL ontology, which covers the structural and behavioral aspects of a pattern analysis algorithm for encoding modal music scores. As explained, the proposed ontology structures the entire music score in addition to its pattern analysis to enable information retrieval and analysis of music score content. The ontology could be extended in the future to support other pattern analysis theories in the music field.

Moreover, we presented the MEI2JSON converter, which covers the transformation of music scores encoded in MEI to JSON format for preprocessing purposes. The proposed converter relies on the MusicPatternOWL ontology to achieve information retrieval and to structure music score content throughout the conversion process. We compared MEI2JSON with a combined approach consisting of two existing converters, Meico and MusicJSON. The experimental results were promising, as our converter was able to outperform the combined converters in terms of data quality assessment.

Our future work will leverage the proposed approaches to efficiently convert music scores to JSON format for upcoming machine learning use cases.

Author Biography

Charbel El Achkar - Antonine University, Faculty of Engineering and Technology, Ticket Lab. Charbel El Achkar is a Postdoctoral Researcher at the Marie and Louis Pasteur University (formerly the University of Franche-Comté). He received his PhD in Computer Science from the University of Bourgogne–Franche-Comté (UBFC), Besançon, France, in 2023. His doctoral research focused on audio transcription, classification, and generation—particularly for music—using audio signal processing and computer vision

techniques. His current research interests include multimodal dataset generation and analysis, computer vision for industrial anomaly detection, and microscopic depth estimation in holographic imaging.

Talar Atéchian - Antonine University, Faculty of Engineering and Technology, Ticket Lab. Talar Atechian is an Associate Professor at the Faculty of Engineering and Technology at Antonine University. She received her PhD in Engineering Sciences from INSA Lyon, France, in 2010. Her current research focuses on the integration of advanced data analytics using applied Artificial Intelligence, with an emphasis on machine learning–based recommender systems, computational modeling for music generation and analysis, and AI-driven education-al technologies.

Publisher's Note

This article is published by Luminous Insights in partnership with Antonine University. Its content was prepared, peer-reviewed, and accepted in accordance with the policies and standards of Antonine University as the original institutional partner. The article has not been published elsewhere, and Luminous Insights is the online publisher of record for this version.

Cite as

El Achkar C. and Atéchian T. (2024). Automation of the Musical Generative Grammar Encoding Process for Modal Monodies. Revue des Traditions Musicales, 18(1), 35–44. 10.51300/RTM-2024-140

References

Abou Mrad, N. (2016). *Éléments de sémiotique modale. Essai d'une grammaire musicale pour les traditions monodiques*. Éditions Geuthner et Éditions de l'Université Antonine.

Ariza, C. (2010). Music21 : A toolkit for computer-aided musicology and symbolic music data. *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, 637–642.

Asmar, M., Atéchian, T., Leblond Martin, S., & Abou Mrad, N. (2018). Traditional modal monodies generative grammar encoding in the Music Encoding Initiative. *Proceedings of the International Conference on Technologies for Music Notation and Representation*, 95–103. https ://doi.org/10.5281/zenodo.1289693

Berndt, A. W., & Hadjakos, A. (2018). Meico : A converter framework for bridging the gap between digital music editions and its applications. *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion (AM'18)*. https ://doi.org/10.1145/3243274.3243282

Bikakis, N., Gioldasis, N., Tsinaraki, C., & Christodoulakis, S. (2009). Querying XML data with SPARQL. In S. S. Bhowmick, J. Küng, & R. Wagner (Éds.), *Database and expert systems applications. DEXA 2009* (Lecture Notes in Computer Science, vol. 5690, pp. 372–386). Springer. https ://doi.org/10.1007/978-3-642-03573-9_32

Breitling, F. (2009). A standard transformation from XML to RDF via XSLT. *Astronomische Nachrichten*, *330*(8), 755–760. https ://doi.org/10.1002/asna.200811233

Cherfi, S., Guillotel, C., Hamdi, F., Rigaux, P., & Travers, N. (2017). Ontology-based annotation of music scores. *Proceedings of the 9th Knowledge Capture Conference (K-CAP '17)*, Article 10, 1–4. https ://doi.org/10.1145/3148011.3148038

El Achkar, C., & Atéchian, T. (2020). Supporting music pattern retrieval and analysis : An ontology-based approach. *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics (WIMS 2020)*, 17–20. https ://doi.org/10.1145/3405962.3405973

Good, M. (2001). MusicXML for notation and analysis. In W. B. Hewlett & E. Selfridge-Field (Éds.), *The virtual score : Representation, retrieval, restoration* (pp. 113–124). MIT Press.

Jones, J., Braga, D., Tertuliano, K., & Kauppinen, T. (2017). MusicOWL : The music score ontology. *Proceedings of the International Conference on Web Intelligence (WI '17)*, 1222–1229. https ://doi.org/10.1145/3106426.3110325

Jsonix. (s.d.). *Jsonix* [Logiciel]. Récupéré de https ://github.com/highsource/jsonix

Kay, M. (2010). *Saxon : The XSLT and XQuery processor* [Logiciel].

Lacoste, D., Sawant, K. P., & Roy, S. (2011). An efficient XML to OWL converter. *Proceedings of the 4th India Software Engineering Conference (ISEC '11)*, 145–154. https ://doi.org/10.1145/1953355.1953376

Lvaro, J. L., & Barros, B. (2010). MusicJSON : A representation for the computer music cloud. *Proceedings of the 7th Sound and Music Computer Conference*, Barcelona.

MIDI 1.0 detailed specification, document version 4.2. (1995). MIDI Manufacturers Association.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn : Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Pezoa, F., Reutter, J. L., Suarez, F., Ugarte, M., & Vrgoč, D. (2016). Foundations of JSON schema. *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*, 263–273. https ://doi.org/10.1145/2872427.2883029

Pugin, L., Zitellini, R., & Roland, P. (2014). Verovio : A library for engraving MEI music notation into SVG. *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 107–112. https ://doi.org/10.5281/zenodo.1417589

RDFLib. (s.d.). *RDFLib* [Logiciel]. Récupéré de https ://github.com/RDFLib/rdflib

Rodrigues, T., Rosa, P., & Cardoso, J. (2008). Moving from syntactic to semantic organizations using JXML2OWL. *Computers in Industry*, *59*(8), 808–819. https ://doi.org/10.1016/j.compind.2008.06.002

Roland, P. (2002). The Music Encoding Initiative (MEI). *Proceedings of the First International Conference on Musical Applications Using XML*, 55–59.

The Music Encoding Initiative. (s.d.). Récupéré de https ://github.com/music-encoding/music-encoding

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. CreateSpace.