TRADITIONAL MODAL MONODIES GENERATIVE GRAMMAR ENCODING IN THE MUSIC ENCODING INITIATIVE

Mark Asmar
Ticket Lab, Faculty of Engineering
Antonine University
mark.asmar.ma@gmail.com

Talar Atechian
Ticket Lab., Faculty of Engineering
Antonine University
talar.atechian@ua.edu.lb

Nidaa Abou Mrad
Research Centre for Musical Traditions
Faculty of Music and Musicology
Antonine University
nidaa.aboumrad@ua.edu.lb

Sylvaine Leblond Martin
Chaire Unesco ITEN
Maison des Sciences de l'Homme Paris Nord
Université Paris 8
sylvaine.leblond-martin@mshparisnord.fr

ABSTRACT

Encoding written music with a textual format is a technology developed for exchanging and analyzing digital music scores. In the literature, many standards exist for this purpose, such as, MusicXML and the Music Encoding Initiative (MEI). The Music Encoding Initiative is a standard developed for encoding music scores in XML. It supports the encoding of different types of notations, such as, Common Music Notation, Neumes Notation, etc. It encodes many features and elements related to musical components such as the pitch name, the octave and the duration of notes. However, for the researchers in musicology, additional information are necessary to enrich the MEI and in order to provide more specific music analysis. In this paper, we target the modal monodies analysis. Thus, we propose to enrich the MEI by appending to its initial schema additional information extracted from the generative grammar of modal monodies. The proposed solution consists of adding a custom module to the MEI containing new elements and attributes. In addition, a new semi-automated analysis component is proposed for the analysis of traditional Modal Monodies of the Middle East and the Mediterranean cultures.

1. INTRODUCTION

An important part of analytical musicology (and analytical ethnomusicology) is dedicated to study the melodic and rhythmic structure of compositions and improvisations from different traditions in the world, and to propose models to explain how it works. Thus, generative grammars were developed and proposed to enrich the analysis of music utterances. A musical generative grammar proposed in the Modal Semiotics theory [6], serves for analyzing traditional modal monodies of the Middle East and the Mediterranean cultures (including medieval European monodic music, as well as Mashriq and Maghreb traditions).

The Music Encoding Initiative encodes music in XML, a textual, yet structured format. An MEI document is an XML document associated to a schema that defines its structure [2]. An XML document would have a schema written in one of the following three languages: the W3C Schema (XSD), RelaxNG (RNG) or Document Type Definition (DTD) [2]. The MEI schema defines and describes the numerous elements existing for music encoding. A group of elements used for the same purpose constitutes a module. Unlike MusicXML, the MEI encodes notations other than the Common Music Notation (CMN) as well as metadata [1]. The Music Encoding Initiative suits our project of encoding the generative grammar of modal monodies according to the Modal Semiotics theory expressed in [6].

The project proposed in [11] and discussed in this paper aims at encoding the generative grammar proposed in [6] of the traditional modal monodies in MEI format. The proposed solution consists of adding a custom module to the MEI. Thus, an algorithm is developed and implemented to extract the grammar out of MEI-encoded music before encoding them back again with its grammar.

The paper structure is as follows: In the next section, a state of the art is presented introducing the added modules to the MEI as well as presenting existing projects concerning music analysis. The third section discusses the proposed solution. The fourth section discusses the experiments conducted to evaluate the proposed solution. Finally, we conclude the paper in the last section.

2. STATE OF THE ART

Several theories-based solutions are presented in the literature for analyzing music scores and partitions. In this section, we discuss solutions developed for analyzing music. In addition, we present contextualized modules added to MEI to enrich the MEI standard schema. The author in [7] developed a user interface for Schenkerian Analysis, aiming to analyze musical scores based on the Schenkerian theory proposed by Heinrich Schenker. In an attempt to overcome the difficulties of its computer implementation, the authors in [8] developed a solution for the Lerdahl and Jackendoff's Generative Theory of Tonal Music. They developed, implemented and tested four different analyzers.

An extension of the Text Encoding Initiative (TEI), a standard developed for encoding texts, made it possible to support the encoding of music within texts. The TEI used to encode texts and music occurring within texts, considering musical pieces or notes as images [4]. By adding the <notatedMusic> element to the TEI, the latter now supports the inclusion of music expressed in MEI, a graphical representation of the music or any other format representing the music [4]. All MEI elements within the <notatedMusic> element are prefixed with "*mei*:", for example <mei:music> [4].

The project described in [9] uses both MEI and TEI, creating a data model and using both the MEI and TEI for encoding of holdings of the Detmold Court Theatre (1825-1875), providing a catalog, which can also be used as a searching tool for specific data [9, 10].

The Solesmes module proposed in [2] captures Solesmes-specific music notation about Gregorian chant. According to [2], the MEI supports the encoding of the neume music notation; however, it lacks some specific features for the Solesmes-neume notation. A new module is proposed, adding new elements and attributes to the MEI schema in order to capture more accurately features related to the Solesmes-neume notation.

A new module proposed in [3], adds layout-related components in MEI, since the latter does not encode information concerning the layout. Using a separate sub-tree, the layout module allows the encoding of information concerning multiple visual representations of the music, while keeping the musical content intact.

However, in order to encode the generative grammar of modal monodies, which is the main goal of this paper, it is necessary to propose a contextualized custom module as an extension of the MEI schema. In addition, a semi-automated algorithm is proposed to implement the analysis process.

3. MODAL SEMIOTICS

The Modal Semiotics theory in [6], describes a generative grammar for modal monodies, related to musical traditions of the Mashriq. This grammar aims at rewriting these modal monodies based on a set of rules, describing mainly some particular modal monodies features such as the *rhythmic* parameter (morphological rewriting) and the *melodic* parameter (rhythmic melodic morphophonological rewriting and modal syntactic rewriting) of the music.

Figure 1. Nuclear Reduction of "Suġītō Qūm fawlōs", a Syriac Maronite Hymn.

According to the phonological component of this theory [6], the final note of a music utterance where the piece ends, helps deciding which notes belong to the alpha or primary modal nucleus and which belong to the beta or secondary modal nucleus (see Figure 1). Compared to the final note (considered as the first degree), even notes are alphas and odd notes are betas. Each "focal note" in the piece placed at the beginning of a syllable is assigned the appropriate symbol, either α or β. This is the "*Syllabic Nuclear Reduction*" [6].

The "*Metasyllabic Nuclear Reduction*" in rhythmic melodic morphophonological rewriting chooses out of the symbols in the previous phases, the ones that are more important. Having the following matrices for the previous musical score:

$$\begin{pmatrix} (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \end{pmatrix} \leftrightarrow \begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$

The first matrix defines the rhythmical structure of the measure, meaning at each quarter note ♩, and each eighth note ♪ have one of the either two symbols, α or β, assigned to it. The second matrix shows that each quarter and eighth note are equal to a single quarter dotted note in the "*Metasyllabic Nuclear Reduction*", taking the symbol of the highest note's duration into consideration, in this case the quarter note.

The next step consists of rewriting the entire music score into rhythmic melodic morphophonological matrices and modal syntactic vectors. The process described is as follows:

$$N\big(E(\mu_1)\big) = N\big(\Lambda(\mu_1)R(\mu_1)\big) = N\big(\Lambda(\mu_1)\big)R(\mu_1) =$$
$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix} \leftrightarrow (\alpha,\alpha,\beta,\alpha)\begin{pmatrix} (|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}) \\ (|\text{tō}|\text{♩}, -|\text{men}|\text{♪}) \\ (|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}) \\ (|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

The matrix containing α and β is derived from the previous "*Metasyllabic Nuclear Reduction"* phase. However, adding the negative sign before notes depends on the multiplier of that note and its associated symbol [6]. For example, |maw|♪ symbol is β while having α as its multiplier, so a minus sign precedes it ($\alpha = -\beta$).

Next, the multiplication of the two matrices takes place to get the following result [6]:

$$\begin{pmatrix} \alpha.\,(|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}), \alpha.\,(|\text{tō}|\text{♩}, -|\text{men}|\text{♪}), \\ \beta.\,(|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}), \alpha.\,(|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

$$\leftrightarrow \begin{pmatrix} (\alpha.\,|\text{leḫ}|\text{♩}, \beta.\,|\text{maw}|\text{♪}), (\alpha.\,|\text{tō}|\text{♩}, \beta.\,|\text{men}|\text{♪}), \\ (\beta.\,|\text{šū}|\text{♩}, \alpha.\,|\text{ro}|\text{♪}), (\alpha.\,|\text{yō}|\text{♩}, \beta.\,|\text{da}'|\text{♪}) \end{pmatrix}$$

Later, notes' pitch names replace alphas and betas to obtain the "*Phonological Realization*":

$$\begin{pmatrix} (C.\,|\text{leḫ}|\text{♩}, D.\,|\text{maw}|\text{♪}), (E.\,|\text{tō}|\text{♩}, F.\,|\text{men}|\text{♪}), \\ (F.\,|\text{šū}|\text{♩}, G.\,|\text{ro}|\text{♪}), \big(E.\,|\text{yō}|\text{♩}, F.\,|\text{da}'|\text{♪}\big) \end{pmatrix}$$

The "*Vector Transcoding*" comes next, which consists of transforming the piece into vectors (primordial prolongative vector $\vec{p}$, suspensive vector $\vec{s}$, questioning vector $\vec{q}$, responsive vector $\vec{r}$) under the following rules, as described in [6]:

$$\overrightarrow{\alpha\alpha} = \vec{p} \qquad \overrightarrow{\beta\beta} = \vec{s} \qquad \overrightarrow{\alpha\beta} = \vec{q} \qquad \overrightarrow{\beta\alpha} = \vec{r}$$

The result for the first measure and the entire piece are as follows respectively:

$$(\vec{p}, \vec{q}, \vec{r}) \qquad (\vec{p}, \vec{q}, \vec{r}, (\vec{p}), \vec{p}, \vec{q}, \vec{r})$$

The so called "*Syntactic Elaboration*" clotures the analysis, by rewriting entirely the music utterance using vectoral decomposition equations, starting with the Fundamental/Original Structure or Primordial Dichotomy equation, as described in [6]:

$$\{\vec{p}\} \rightarrow \{[\vec{q}] + [\vec{r}]\}$$

4. PROPOSED SOLUTION

In order to attend our goal, the solution consists of adding a new custom module, named "grammar" to the MEI schema. In addition to the semi-automated algorithm for music scores analysis.

4.1 The Schema Extension

Similar to TEI, MEI schema is extensible. It allows enriching the encoding process by contextualized custom modules. Thus, developers can generate their own custom schemas out of the initial MEI schema [2]. First, an XML document describes the expected custom output schema, then using the TEI stylesheets; the latter use the XML file and the MEI schema to generate a schema file describing the expected MEI schema. Referring to this technical approach, the new custom module so called "*grammar*" is generated, for encoding generative grammar of Modal Monodies within MEI. The new elements and attributes added to the schema are present below in Figure 2.

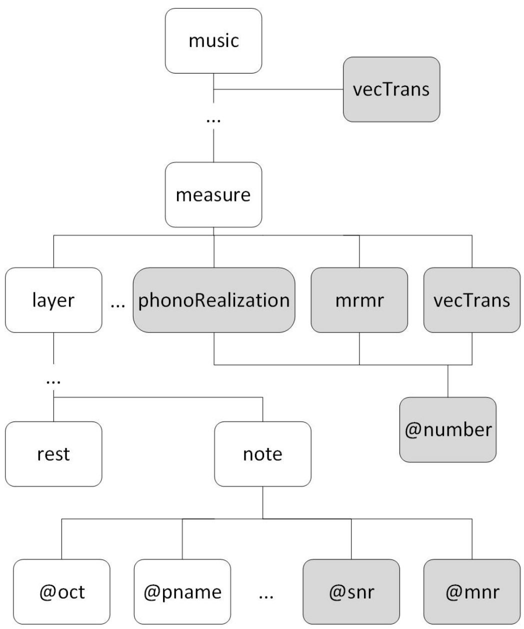

Figure 2. Newly added elements and attributes to the MEI schema.

Table 1 describes the elements and attributes of the "*grammar*" module.

Attributes	
Attribute	**Description**
snr	Attribute for the note element. Used for the "*Syllabic Nuclear Reduction*", it may contain only two values α and β.
mnr	Attribute for the note element. It may contain "yes" or "no" values, describing whether the note is taken into consideration in the "*Metasyllabic Nuclear Reduction*" or not.
number	Serves as an identifier for the <phonoRealization>, <mrmr> and <vecTrans> elements in the case of considering repetitions, reflecting the index of the measure in the analysis before and after the repetition.
Elements	
Element	**Description**
mrmr	Child of the <measure> element. Destined for the encoding of the "*Morphophonological Rhythmic and Melodic Rewriting*"; it contains matrices and mathematical equations.
phonoRealization	Child of the <measure> element. Used to encode the "*Phonological Realization*" phase of the analysis, just like *mrmr* it contains matrices and equations as well.
vecTrans	Child of both the <measure> and <music> elements. It serves as the element for the encoding of the "*Vector Transcoding*" phase, and it contains vectors.

Table 1. Description of the components of the "grammar" module.

All elements and attributes present in Table 1 contain mathematical expressions, symbols or expressions, except for the *mnr* and *number* attributes, so having a textual format for representing equations within XML is mandatory. The *snr* attribute has only two values, "\alpha" or "\beta", while the three elements contain equations including matrices and vectors, making the TeX language suitable, similar to the TEI encoding of mathematical expressions [5]. The TeX package, "lilyglyps" a package for displaying "Lilypond" music symbols, used for the representation of notes and rests in the matrices during the encoding process.[1]

[1] https://ctan.org/pkg/lilyglyphs

4.2 The Semi-Automated Music Analysis Algorithm

The semi-automated music analysis algorithm is presented in this section. It consists of implementing the music analysis proposed in [6], and to encode the result in the "*grammar*" module. However, the phases stating from the "*Syllabic Nuclear Reduction*" until the "*Vector Transcoding*" were done, leaving the last phase, "*Syntactic Elaboration*", unimplemented.

The algorithm takes as an input a score of music encoded in MEI, and returns an updated MEI document containing the "*grammar*" elements and the attributes as results of the analysis. The analysis process proceeds in a sequential manner, for each measure in the analyzed music score, all analysis phases are applied; when completed, the next measure is analyzed and so on.

The same process executes when the analysis includes repetitions except that prior to the analysis repeated measures is mandatory. In the case, of multiple verses in each processing of the measure the appropriate verse is taken into account.

4.2.1 Syllabic Nuclear Reduction

This phase consists of assigning for each note in the music score α and β symbols. First, the lowest and the highest notes of the music score are identified.
In addition, the algorithm identifies the final note of the music score and identified as alpha note (α). According to this reference note, the alpha (α) and beta (β) notes of the music score are identified.

Descending from the final note towards the lowest note, each even note after the final note is an alpha while odd notes are betas, for example, if the fourth octave C note (C4) is the final note, and the third octave A note (A3) is the lowest, C4 is α, B3 is β and A3 is α. The process stops when the algorithm reaches the lowest note in the piece. The second part constitutes of a similar process, starting from the final note and moving towards the highest note, even notes are alpha and odd notes are betas as well, for example, if C4 is the final note, and F4 is the highest, C4 is α, D4 is β, E4 is α and F4 is β. Next, the algorithm checks the music given as input for syllables, if they exist then the notes residing at the beginning of each syllable are assigned to their corresponding symbol from the aforementioned array. If no syllables exist, then a calculation of the nuclear reduction based on a user given matrix takes place. Having the following piece of music [6]:

Figure 3. Dūlāb Rāst.

Moreover, the following matrix (based on the percussive *waḥda* cycle *dum*♩, *tak* ♪, *tak* ♪):

$$\begin{pmatrix} ♩ \\ (♪,♪) \end{pmatrix}$$

The analysis algorithm tries to find out matches between the notes within the matrix, represented by the following values [[0.25], [0.125, 0.125]], and those within every measure in the music score. The above matrix describes each measure, meaning that every measure contains a quarter note and two eighth notes, or their equivalents; for example, two eighth notes instead of a quarter note. By processing each measure sequentially, each note's duration[2]

$$\frac{1}{n} \tag{1}$$

(a quarter note would have a duration of $\frac{1}{4}$, n retrieved from the note's *dur* attribute) is checked with the first element of the matrix. Considering that element is d, the *snr* attribute is added in case the note's duration is equal to the duration of the latter, while the result of (1) is added to a sum in the other case. The first note of the sum is the note that has the *snr* attribute added to it when the sum is equal to d, meaning that a series of notes equal to d were found, for example, finding two eighth notes while searching for a quarter note. When an element of the matrix is found, the algorithm proceeds to the next one. Dotted notes are calculated using their corresponding equivalents, for example, a ♩. is replaced by a quarter note and an eighth note ♩. = ♩ + ♪ . The equivalents' durations are useful for calculating the durations using the aforementioned formula. Figure 4 shows the result for Figure 3, as follows:

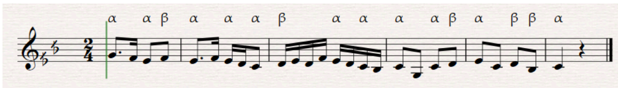

Figure 4. Dūlāb Rāst after SNR phase.

The analysis of a syllabled music score (see Figure 5), returns the result as illustrated in Figure 6.

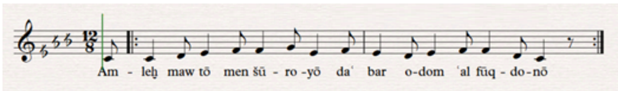

Figure 5. Suġītō Qūm fawlōs.

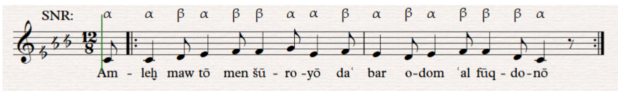

Figure 6. SNR phase result for Suġītō Qūm fawlōs.

The outcome of the "*Syllabic Nuclear Reduction*" phase introduces new elements in MEI encoding document as follows:

```
<note xml:id="m-40" dur="4" dur.ges="256p"
oct="4" pname="c" pnum="48" stem.dir="up"
snr="\alpha">
        <verse n="1">
                <syl wordpos="t">leḫ</syl>
        </verse>
</note>
```

[2] A quarter note would have a duration of four in MEI for example, however in this paper MEI durations are inversed 1.

In the element <note>, specific attributes and elements are defined as follows: the duration (*dur*) of the note, the pitch name (*pname*), the octave (*oct*), while having elements like <verse> and <syl> describing the verses and the syllables respectively.

In addition, new attribute (*snr*) is added to the schema, representing the assigned nucleus of a note in the "*Syllabic Nuclear Reduction*".

4.2.2 Metasyllabic Nuclear Reduction

The "*Metasyllabic Nuclear Reduction*" phase initiates after the completion of the previous phase. It needs as an input an initial matrix manually provided at the beginning of the analysis [6]. The following matrix is an example for the score in Figure 5.

$$SNR\ Matrix\!: \begin{pmatrix} (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \\ (\text{♩},\text{♪}) \end{pmatrix} \leftrightarrow MNR\ Matrix\!: \begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$

The analysis algorithm receives the matrices as arrays. A numerical value is assigned to each note as follows: [[0.25, 0.125], [0.25, 0.125], [0.25, 0.125], [0.25, 0.125]] and [0.375, 0.375, 0.375, 0.375] respectively. These values in the array stand for the duration of the notes in the matrix, and the arrays within the initial array represent the rows within the matrix.

For each row in the matrix, the corresponding note containing a *snr* attribute for the highest duration in that row is chosen in the "*Metasyllabic Nuclear Reduction*". The algorithm adds the *mnr* attribute to that note with a value of "yes". As illustrated in the music score of the Figure 6, the first row of its "SNR Matrix" contains a quarter and an eighth note. The quarter note being the highest, the first note in the first measure, will have the *mnr* attribute set to "yes", as for the second row in the matrix and the third note in the piece. The same process is applied for each row in the matrix.

This process is the same for every measure in the piece. The music score shown in Figure 7 contains the MNR and SNR.

Figure 7. MNR phase result for Suġītō Qūm fawlōs.

As an example, the *mnr* attribute of the element <note> is represented as follows:

```
<note xml:id="m-40" dur="4" dur.ges="256p"
oct="4" pname="c" pnum="48" stem.dir="up"
snr="\alpha" mnr="yes">
        <verse n="1">
                <syl wordpos="t">leḫ</syl>
        </verse>
</note>
```

4.2.3 Morphophonological Rhythmic and Melodic Rewriting

After adding both the *snr* and *mnr* attributes, the process of generating matrices and the equations is necessary for the "*Morphophonological Rhythmic and Melodic Rewriting*".

We consider δ the matrix of *alphas* and *betas*. The notes of a measure containing the *snr* and *mnr* attributes having the latter set to "yes", will have their *snr* attribute values appended to δ.

$$\delta = (\alpha,\alpha,\beta,\alpha)$$

The above example represents the matrix generated for the first measure of the music score shown in Figure 6. This matrix is then, multiplied by the matrix given for the "*Metasyllabic Nuclear Reduction*", shown in Figure 6. The result is represented as follows:

$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} \text{♩.} \\ \text{♩.} \\ \text{♩.} \\ \text{♩.} \end{pmatrix}$$

Next, the δ multiplies the "SNR Matrix" shown in the section 4.4.2. However, before the multiplication, if the matrix contains more than one element in any of its rows, then its corresponding symbol within the δ matrix is compared to its *snr* attribute value. If they are different then a negative sign precedes the note. In the case of a syllabled music score, the corresponding syllables precede the notes as well.

$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} (|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}) \\ (|\text{tō}|\text{♩}, -|\text{men}|\text{♪}) \\ (|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}) \\ (|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

Considering −|maw| ♪, the negative sign indicates that the note's equivalent within the music is β while its multiplier is α in the δ matrix.

The multiplication of the two matrices continues by multiplying each element in the δ matrix by its corresponding row in the second matrix. Resulting the following output:

$$\begin{pmatrix} \alpha.\,(|\text{leḫ}|\text{♩}, -|\text{maw}|\text{♪}), \alpha.\,(|\text{tō}|\text{♩}, -|\text{men}|\text{♪}), \\ \beta.\,(|\text{šū}|\text{♩}, -|\text{ro}|\text{♪}), \alpha.\,(|\text{yō}|\text{♩}, -|\text{da}'|\text{♪}) \end{pmatrix}$$

The final step of this entire phase[3] is the process of multiplying the symbols α and β with the elements present in each row, directly. If a minus precedes the element within the row, -its *snr* attribute is different from the multiplier- the multiplier changes, α turns into β and vice versa.

$$\begin{pmatrix} (\alpha.\,|\text{leḫ}|\text{♩}, \beta.\,|\text{maw}|\text{♪}), (\alpha.\,|\text{tō}|\text{♩}, \beta.\,|\text{men}|\text{♪}), \\ (\beta.\,|\text{šū}|\text{♩}, \alpha.\,|\text{ro}|\text{♪}), (\alpha.\,|\text{yō}|\text{♩}, \beta.\,|\text{da}'|\text{♪}) \end{pmatrix}$$

The matrices shown in this section are expressed in TeX format as added in the element named <mrmr>, child of the <measure> element, as a TeX string, as shown below:

[3] Notes of the matrix are replaced by rests if rests exist at the end of a measure

```
<mrmr>
\begin{pmatrix}
\alpha , \alpha , \beta , \alpha
\end{pmatrix}
\begin{pmatrix}
\quarterNoteDotted \\ \quarterNoteDotted \\
\quarterNoteDotted \\ \quarterNoteDotted
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha , \alpha , \beta , \alpha
\end{pmatrix}
\begin{pmatrix}
( |leẖ| \quarterNote,-|maw| \eighthNote ) \\
( |tō| \quarterNote,-|men| \eighthNote ) \\
( |šū| \quarterNote,-|ro| \eighthNote ) \\
( |yō| \quarterNote,-|da'| \eighthNote )
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha.( |leẖ| \quarterNote,-|maw| \eighthNote
),
\alpha.( |tō| \quarterNote,-|men| \eighthNote ),
\beta.( |šū| \quarterNote,-|ro| \eighthNote ),
\alpha.( |yō| \quarterNote,-|da'| \eighthNote )
\end{pmatrix}
\\
\\
\begin{pmatrix}
( \alpha.|leẖ| \quarterNote,\beta.|maw| \eighth-
Note ) ,
( \alpha.|tō| \quarterNote,\beta.|men| \eighth-
Note ) ,
( \beta.|šū| \quarterNote,\alpha.|ro| \eighth-
Note ) ,
( \alpha.|yō| \quarterNote,\beta.|da'| \eighth-
Note )
\end{pmatrix}
</mrmr>
```

If repetitions are considered in the analysis, the <mrmr> element is added twice with different values for the *number* attribute. For the first measure of the score shown in Figure 6, the result is as the following:

```
<mrmr number="1">
...
</mrmr>
<mrmr number="3">
...
</mrmr>
```

4.2.4 Phonological Realization

This phase consists of multiplying each element of the "SNR Matrix" with its corresponding note's pitch name. In the analysis algorithm, this step is part of the "*Morpho-phonological Rhythmic and Melodic Rewriting*". Repeating the final step performed in the previous phase, replacing *alphas* and *betas* by the corresponding pitch names of notes. A string in the TeX syntax expresses the entire equation, encoded as the value of the <phonoRealization> MEI element that is a child element of the <measure> element.

The "*Phonological Realization*" of the first measure of the music score shown in Figure 7 is represented as follows:

$$\begin{pmatrix} (C.|\text{leẖ}|♩, D.|\text{maw}|♪), (E.|\text{tō}|♩, F.|\text{men}|♪), \\ (F.|\text{šū}|♩, G.|\text{ro}|♪), (E.|\text{yō}|♩, F.|\text{da'}|♪) \end{pmatrix}$$

```
<phonoRealization>
\begin{pmatrix}
( C.|leẖ|\quarterNote , D.|maw|\eighthNote ) ,
( E.|tō|\quarterNote , F.|men|\eighthNote ) ,
( F.|šū|\quarterNote , G.|ro|\eighthNote ) ,
( E.|yō|\quarterNote , F.|da'|\eighthNote )
\end{pmatrix}
</phonoRealization>
```

If repetitions are considered in the analysis, the <phonoRealization> element is added twice with different values for the *number* attribute, similarly to the <mrmr> element. For the first measure of the score shown in Figure 7, the result is as the following:

```
<phonoRealization number="1">
...
</phonoRealization>
<phonoRealization number="3">
...
</phonoRealization>
```

4.2.5 Vector Transcoding

The last step of the entire analysis algorithm consists of generating vectors out of the musical analysis. Using the values of the *snr* attributes of notes taken into consideration in the "*Metasyllabic Nuclear Reduction*", each two consecutive values are equal to a vector based on the following rules [6]:

$$\overrightarrow{\alpha\alpha} = \vec{p} \qquad \overrightarrow{\beta\beta} = \vec{s} \qquad \overrightarrow{\alpha\beta} = \vec{q} \qquad \overrightarrow{\beta\alpha} = \vec{r}$$

Two consecutive notes containing the *snr* attribute set, with the *mnr* attribute equal to "yes", are replaced by their corresponding vector based on their *snr* attribute values. Vectors generation takes place on both, a per measure basis and for the entire music. In the second case, which executes when all measures are analyzed, a vector generated by two symbols that belong to different measures, an opening parenthesis and a closing parenthesis precedes and succeeds the vector respectively. At last, the <vecTrans> element is added to both <music> and <measure> elements within MEI.

The following represents the result of the *Vector Transcoding* phase for both the first measure and the entire score shown in Figure 7.

$$(\vec{p}, \vec{q}, \vec{r})$$

```
<vecTrans>
\begin{pmatrix}
\vec{p} , \vec{q} , \vec{r}
\end{pmatrix}
</vecTrans>
```

$$(\vec{p}, \vec{q}, \vec{r}, (\vec{p}), \vec{p}, \vec{q}, \vec{r})$$

```
<vecTrans>
\begin{pmatrix}
\vec{p} , \vec{q} , \vec{r} ,
(\vec{p}) ,
\vec{p} , \vec{q} , \vec{r}
\end{pmatrix}
</vecTrans>
```

Like both the <mrmr> and <phonoRealization> element occurs twice with different values for the *number* attribute. For the first measure of the score shown in Figure 7, the result is as the following:

```
<vecTrans number="1">
...
</vecTrans>
<vecTrans number="3">
...
</vecTrans>
```

4.2.6 Anacrusis

One important aspect of the analysis is dealing with the anacrusis. A "*Syllabic Nuclear Reduction*" occurs on the measure of the anacrusis if it exists, however, unlike the process described earlier; this process applies in the inverse. The process starts by finding the notes from the end of the matrix and backwards. For the example shown in Figure 5, and using the matrix shown in the "*Metasyllabic Nuclear Reduction*" section 3.3.2, an eighth note is found at the end of the matrix. This note left as it is while rests replace the other notes in the matrix for the "*Morpho-phonological Rhythmic and Melodic Rewriting*". The matrix that contains *alphas* and *betas* is retrieved from the last measure in case of the absence of repetitions, or from the measure that returns to the beginning in the other case. The same process described earlier for both the "*Morpho-phonological Rhythmic and Melodic Rewriting*" and "*Phonological Realization*" executes, while replacing absent notes with rests, and replacing pitches in the latter phase by underscores. The result of the anacrusis analysis of the piece shown in Figure 7 would be as following in both Figure 8 and Figure 9:

$$N(E(\mu_0)) = N(\Lambda(\mu_0)R(\mu_0)) = N(\Lambda(\mu_0))R(\mu_0) =$$

$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} ♩. \\ ♩. \\ ♩. \\ ♩. \end{pmatrix}$$

$$(\alpha,\alpha,\beta,\alpha)\begin{pmatrix} (𝄽, 𝄾) \\ (𝄽, 𝄾) \\ (𝄽, 𝄾) \\ (𝄽, |\mathrm{Am}|♪) \end{pmatrix}$$

$$(\alpha.(𝄽, 𝄾), \alpha.(𝄽, 𝄾), \beta.(𝄽, 𝄾), \alpha.(𝄽, |\mathrm{Am}|♪))$$

Figure 8. MRMR for the anacrusis.

$$((\alpha.𝄽, \beta.𝄾), (\alpha.𝄽, \beta.𝄾), (\beta.𝄽, \beta.𝄾), (\alpha.𝄽, \alpha.|\mathrm{Am}|♪))$$

$$E(\mu_0) = ((\_.𝄽, \_.𝄾), (\_.𝄽, \_.𝄾), (\_.𝄽, \_.𝄾), (\_.𝄽, C.|\mathrm{Am}|♪))$$

Figure 9. MRMR and PR for the anacrusis.

```
<mrmr>
\begin{pmatrix}
\alpha, \alpha, \beta, \alpha
\end{pmatrix}
\begin{pmatrix}
\quarterNoteDotted \\ \quarterNoteDotted \\
\quarterNoteDotted \\ \quarterNoteDotted
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha, \alpha, \beta, \alpha
\end{pmatrix}
\begin{pmatrix}
(\crotchetRest, \quaverRest) \\
(\crotchetRest, \quaverRest) \\
(\crotchetRest, \quaverRest) \\
(\crotchetRest, |Am|\eighthNote)
\end{pmatrix}
\\
\\
\begin{pmatrix}
\alpha.(\crotchetRest, \quaverRest) ,
\alpha.(\crotchetRest, \quaverRest) ,
\beta.(\crotchetRest, \quaverRest) ,
\alpha.(\crotchetRest, |Am|\eighthNote)
\end{pmatrix}
\\
\\
\begin{pmatrix}
(\alpha.\crotchetRest, \beta.\quaverRest) ,
(\alpha.\crotchetRest, \beta.\quaverRest) ,
(\beta.\crotchetRest, \beta.\quaverRest) ,
(\alpha.\crotchetRest, \alpha.|Am|\eighthNote)
\end{pmatrix}
</mrmr>
<phonoRealization>
\begin{pmatrix}
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, \_.\quaverRest) ,
(\_.\crotchetRest, C.|Am|\eighthNote)
\end{pmatrix}
</phonoRealization>
```

5. EXPERIMENTS

5.1 Analysis Algorithm Evaluation

The algorithm discussed in this paper and implemented in JavaScript using NodeJS, contains two modules, one of them processing repetitions. It provides the possibility to choose whether to consider repetitions in the analysis or not alongside providing the necessary matrices for the process. Depending on the choice, the appropriate module executes. The algorithm accepts as an input an MEI document, and outputs another MEI document containing the "*grammar*" module elements and attributes with their appropriate values alongside a PDF file containing the piece rendered using Verovio [12] and SVG processing for placing *alphas* and *betas* above notes, with the entire analysis expressed in terms of mathematical expressions.

We chose twelve music scores from [6] for testing and evaluating the correctness of the implemented analysis algorithm. The correctness verification is the most important criteria monitored and evaluated during the testing procedure. The correctness measurement is phase based, meaning an analysis is not entirely wrong if an error occurs at only one phase. However, if an error exists in one phase that may affect the next one, the latter is considered correct if its output is correct considering the error caused by a previous phase.

The algorithm analyzes and encodes all measures. However, since the algorithm cannot yet analyze correctly other than the first measure when encountering a strophic song, an exception is made, evaluating only the first measure for strophic songs. While in the other case, the evaluation considered all measures. The phases are represented by P1, P2, P3, P4 and P5, which represent the "*Syllabic Nuclear Reduction*", "*Metasyllabic Nuclear Reduction*", "*Morpho-phonological Rhythmic and Melodic Rewriting*", "*Phonological Realization*" and "*Vector Transcoding*" respectively. The results for the conducted experiments are present in Table 2.

Instrumental Music					
Music	**P1**	**P2**	**P3**	**P4**	**P5**
Repetition Included					
Dārij Ḥijāz	✓	✓	✓	✓	✓
Repetition Free					
Dūlāb Rāst	✓	✓	✓	✓	✓
Syllabled Music					
Music	**P1**	**P2**	**P3**	**P4**	**P5**
Repetition Included					
Huwwāra	✓	✓	✓	✓	✓
Jibnā l-kibbī wi-l-ḥinnā	✓	✓	✓	✓	✓
Jibnā l-‘arūs uw-jīnā	✓	×	✓	✓	✓
Daḥṭō lō neḥtē	✓	✓	✓	✓	✓
Hymn A02	✓	✓	✓	✓	✓
Suġītō Qūm fawlōs	✓	✓	✓	✓	✓
Bo’ūtō dmor afrēm	✓	✓	✓	✓	✓
Repetition Free					
Dal‘ōnā	✓	✓	✓	✓	✓
Yā šamsi ḍallik šāriqā	✓	✓	✓	✓	✓
Yā šamsi ḍallik šāriqā	✓	✓	✓	✓	✓

Table 2. Results of the performed tests.

5.2 Discussion

Evaluating only the first measure for strophic songs, eleven out of twelve pieces analyses were correct, representing a 92% correctness.

The results presented in the section 5.1, show that the algorithm is able to analyze the majority of modal monodies. In the case of strophic songs, the evaluation takes into account the first measure only; otherwise, all measures were evaluated. The analyses of eleven pieces were completely correct, while only the analysis of “Jibnā l-‘arūs uw-jīnā” was not. This is due to an unsupported case that caused an incorrect output in the second phase, the “*Metasyllabic Nuclear Reduction*” phase, choosing α instead of a β. This is due to the lack of support for some cases rarely present in some pieces. Other cases exist as well, that are not supported and yet to be done.

6. CONCLUSION AND FUTURE WORK

This paper discussed a solution proposed for encoding traditional modal monodies generative grammar in MEI. The solution consists of adding a new custom module to the initial MEI schema, alongside developing an analysis algorithm for the extraction and encoding of the generative grammar. Three new elements and three attributes were added to the MEI schema by creating the grammar module. The implemented algorithm, analyzes the musical scores as per described in [6]. All mathematical expressions and matrices were expressed in TeX, making it easier to understand and render equations, while using the “lilyglyphs” TeX package in order to represent notes like quarter notes ♩ and eighth notes ♪ . The conducted experiments tested the correctness of the implemented analysis algorithm and the results are evaluated satisfactory.

The algorithm helps performing the analysis in an automatic and time saving manner. The results provided by the output of the algorithm may also be used to generate new modal monodies.

As future work, improvements are expected to enhance the analysis algorithm in order to implement specific identified cases in the modal semiotics theory [6] not still supported in the analysis algorithm.

In addition, one of feature is to import MusicXML documents and analyze them as MEI files. Finally, it is expected to automatically identify the input matrices needed for both the “*Syllabic Nuclear Reduction*” for instrumental scores and the “*Metasyllabic Nuclear Reduction*” phase.

7. REFERENCES

[1] “An introduction to MEI” [Online]. Available: http://music-encoding.org/about/

[2] A. Hankinson, P. Roland, and I. Fujinaga, “The Music Encoding Initiative as a Document-Encoding Framework.,” in *International Society for Music Information Retrieval Conference*, Miami, USA, 2011, pp. 293–298.

[3] L. Pugin, J. Kepper, P. Roland, M. Hartwig, and A. Hankinson, “Separating Presentation and Content in MEI.,” in *International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012, pp. 505–510.

[4] TEI Music SIG, “TEI with Music Notation.” [Online]. Available: http://www.tei-c.org/SIG/Music/twm/.

[5] The Text Encoding Initiative, “14 Tables, Formulæ, Graphics and Notated Music,” *The Text Encoding Initiative Guidelines*. [Online]. Available: http://www.tei-c.org/release/doc/tei-p5-doc/en/html/FT.html.

[6] N. Abou Mrad, *Eléments de sémiotique modale : Essai d’une grammaire musicale pour les traditions monodiques*, Paris and Hadat/Baabda: Editions Geuthner and Editions de l’Université Antonine, 2016.

[7] A. Marsden, “Software for Schenkerian Analysis,” in *Proceedings of the International Computer Music Conference*, Huddersfield, UK, 2011.

[8] Hamanaka M., K. Hirata, and Tojo S., “Implementing Methods for Analysing Music Based on Lerdahl and Jackendoff’s Generative Theory of Tonal Music,” in D. Meredith D (Ed.) *Computational Music Analysis*. Springer, 2016.

[9] I. Capelle and K. Richts, “Die Welt des Detmolder Hoftheaters – erschlossen mit MEI und TEI,” *Bibliotheksdienst*, vol. 50, no. 2, pp. 199–209, 2016.

[10] “Das Modell | Detmolder Hoftheater.” [Online]. Available: http://hoftheater-detmold.de/?page_id=1095.

[11] S. Leblond Martin (Ed.), *Musiques orales, notations musicales et encodages numériques / Oral music, music notation and digital encoding*. Paris: Les Éditions de l’Immatériel, 2016.

[12] L. Pugin, R. Zitellini, and P. Roland, “Verovio: A library for Engraving MEI Music Notation into SVG.,” in *International Society for Music Information Retrieval Conference*, Taipei, Taiwan, 2014, pp. 107–112.