Noname manuscript No.
(will be inserted by the editor)

An MEI-based standard encoding for hierarchical music analyses

David Rizo · **Alan Marsden**

Received: date / Accepted: date

Abstract We propose a standard representation for hierarchical musical analyses as an extension to the Music Encoding Initiative (MEI) representation for music. Analyses of music need to be represented in digital form for the same reasons as music: preservation, sharing of data, data linking, and digital processing. Systems exist for representing sequential information, but many music analyses are hierarchical, whether represented explicitly in trees or graphs or not. Features of MEI allow the representation of an analysis to be directly associated with the elements of the music analyzed. MEI's basis in TEI (Text Encoding Initiative), allows us to design a scheme which reuses some of the elements of TEI for the representation of trees and graphs. In order to capture both the information specific to a type of music analysis and the underlying form of an analysis as a tree or graph, we propose related "semantic" encodings, which capture the detailed information, and generic "non-semantic" encodings which expose the tree or graph structure. We illustrate this with examples of representations of a range of different kinds of analysis.

Keywords Encodings · standards · music analysis · music representations

This work was partially supported by the Spanish Ministerio de Economía, Industria y Competitividad through HispaMus project (TIN2017-86576-R).

David Rizo
Department of Software and Computing Systems, University of Alicante
Instituto Superior de Enseñanzas Artísticas de la Comunidad Valenciana
E-mail: drizo@dlsi.ua.es

Alan Marsden
Lancaster Institute for the Contemporary Arts, Lancaster University
E-mail: a.marsden@lancaster.ac.uk

1 Introduction: desirability of an analysis encoding standard

Musical scholarship in the digital age requires not just digital representations of music, but also digital representations of data associated with that music. For some time, it has been recognised that standards are required for recording bibliographic and related metadata about music.[1] More recently, well defined and controlled vocabularies [21] and ontologies [27, 4][2] for musical metadata have been established. However, commentary about pieces of music and analyses go beyond what can properly be called metadata: they can be quite extensive, they can constitute original works in their own right, but still require association with specific pieces of music and often specific locations within those pieces. Analyses are commonly referred to in academic writing about pieces of music, and the knowledge which they embody has other musical uses also. For example, to segment a piece of music at an appropriate place requires knowledge of the phrase structure. To present a snippet which is properly representative of a piece requires knowing where the theme occurs. (It might not be at the beginning!)

As documents progressively move into or are created in the digital domain [33], it is therefore important that there are proper ways of representing and preserving analytical musical knowledge. There are well established means of dealing with text (e.g. Voyant Tools[3]), and sophisticated digital representations of music are being adopted, for example the Music Encoding Initiative described below [10]. Analyses are often represented in diagrams. These can be scanned and stored as images, of course, but this does not necessar-

[1] See for example the Variations projects at the Indiana University, `http://www.dlib.indiana.edu/projects/variations3/`

[2] `http://musicontology.com/`

[3] `https://voyant-tools.org`

ily preserve the knowledge they embody. An image can be separated from the text which explains it, the knowledge of how to interpret its notation can become obsolete, or a scan might simply be at too low a resolution for the details of the diagram to be readable. (Both authors have experienced this situation when using repositories of scanned printed journal articles or books – see figures of analyses containing big trees in [19]). If, on the other hand, a representation uses a scheme which has a precise and formal description, it remains readable so long as the language used for its definition remains current. Furthermore, assistive technologies are able to use such representations to render information in a way accessible to those with visual impairments and on different devices using responsive technologies.

By "reading" we mean not only reading by humans, of course. To be properly usable in the digital domain, a representation of an analysis needs also to be readable by software. It needs to be searchable and, with the growth of the semantic web, it needs also to be linkable [34]. After reading an analysis, it can be useful also for an application to be able to "play" or render it either visually or acoustically. In the field of music information retrieval (MIR), sets of machine-readable data are increasingly important for research. Datasets containing harmonic analyses have been widely used in MIR where modern machine-learning methods require large quantities of data. Commonly used examples include chord-labelling data for Beatles songs [11][4] and other popular music [3].[5] In the symbolic domain, harmonic analyses of Bach's chorales are also commonly used.[6] The representations which such datasets use, though usually well defined, are *ad hoc* and not always related to established standards. Such standards would significant improve the reusability of data and the scope for widespread testing of research results.

The situation is particularly complicated for kinds of analysis which are essentially hierarchical, such as reductions using Schenkerian theory [31] or that of Lerdahl and Jackendoff [19]. These are two highly influential methods of music analysis, and while they strictly cover only a small portion of analytical commentary on music, it is common to find that other analyses have a similar (though often less well developed) hierarchical aspect. Figure 1 for example, taken from [6], points out a recurring pattern in this music. It first extracts the four measures containing the pattern from the rest of the score (as indicated by the square brackets), and then identifies two smaller units within those four bars (indicated by the angled brackets). Across this runs another level of extraction which identifies two pairs of notes in just the upper and lower voices. These are labelled with their scale degrees (1, 7, 4, 3 and 1, 2, 7, 1, indicating the first, seventh, etc., note of the scale) and arrows indicate the opposite directions of movement. Other labels are added to indicate the chord progressions of the two sub-units. There are several layers to this analysis, therefore, which align with the details of the score in different ways.

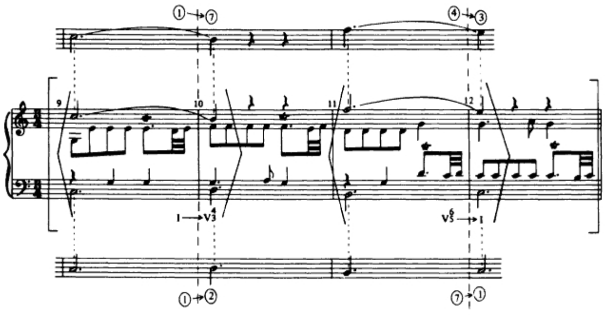


Fig. 1 Example of an analytical diagram with implicit hierarchical elements; from Gjerdingen [6, Ex. 4-11, p.65].

While schemes such as Humdrum[7] already exist which are good for representing sequential information to be aligned with score information, and *ad hoc* sequential representations using an underlying scheme such as Humdrum can be easy to define and use, this is not true for representing the information in hierarchical analyses. Some datasets containing hierarchical musical analyses do exist (e.g., analyses in the style of Lerdahl and Jackendoff [8] and Schenkerian analyses from text books and the like [17]) but their representation schemes are idiosyncratic and not easy to associate with a representation of the music analyzed. In this paper we therefore propose a means of representing hierarchical music-analytical information, i.e., information naturally represented in trees or graphs, strongly related to a standard for representing musical scores, allowing the tight association of the analytical information and the information in the score.

In section 2 we discuss hierarchical analysis further, to illustrate the issues and more precisely define what we mean. In section 3 we describe some existing representations of hierarchical analyses before describing the required features of a standard scheme in section 4. We describe the reasons for using the Music Encoding Initiative (MEI) standard in section 5 and present our proposals for an extension to this standard for hierarchical music analyses.

2 Hierarchical analysis and representation

Hierarchies have had an important role both in musical analysis and, more recently, in representations of music in tasks

[4] `http://isophonics.net/content/reference-annotations-beatles`

[5] `http://ddmal.music.mcgill.ca/research/billboard`

[6] `https://github.com/craigsapp/bach-370-chorales`

[7] `http://www.humdrum.org/` and `https://csml.som.ohio-state.edu/Humdrum/`

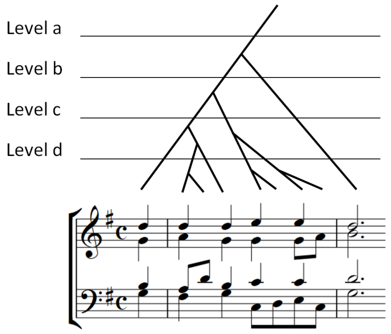
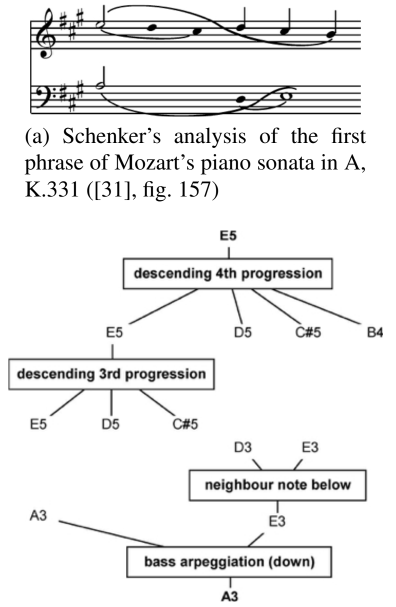


Fig. 2 Generative theory of tonal music (GTTM) reduction at different levels



(a) Schenker's analysis of the first phrase of Mozart's piano sonata in A, K.331 ([31], fig. 157)

(b) Tree representation derived from Schenker's analysis

Fig. 3 Marsden's tree representation Schenker's analysis as proposed in [23] (figures 5 and 6).

such as computer assisted composition or music information retrieval. By "hierarchical analysis" or "hierarchical representation", we mean an analysis or representation which presents the structure of the music on different levels, in which higher levels correspond to larger spans of the music. Relations between different elements are shown, but only between neighbouring elements, either above or below in the hierarchy, or side-to-side from one "branch" to another. An analysis which is based on the relationships of distant segments of music, such as a "paradigmatic analysis" in the style of Nattiez [25], is not hierarchical according to our definition.[8] While hierarchical analyses and representations are often presented as trees, it is not essential for them to be so, as will become clear from the examples below.

The two most famously hierarchical kinds of music analysis are those of Schenker and of Lerdahl and Jackendoff. The well known Generative Theory of Tonal Music (GTTM) [19] proposes several viewpoints for analyzing tonal music based on generative linguistic theory, i.e., modeling analysis using formal grammars such as those used to identify the *parts of speech* (i.e. noun, verb, etc.) in a written sentence (see Figures 2, 10, 11, and 13). Although the theory is presented as a set of rules, and the results are presented in tree diagrams, the analyses are not so systematic as this might suggest. For discussion of some of the difficulties concerning polyphony and uncertainties over what information is maintained from one level to the next, see [23]. Other issues are discussed in [8] which also demonstrates some of the difficulties over systematic treatment of Lerdahl and Jackendoff's "preference rules."

While the analyses of Heinrich Schenker were not presented in trees, levels of hierarchy are explicit, and the analyses can (with the loss of some information) be converted to trees, as demonstrated in [23] (see Figure 3). Further systematisations of Schenkerian analysis are found in [22, 24] and [17]. The latter presents an analysis not as a tree but as a *Maximal outerplanar graph* (MOP), a representation for Schenkerian analyses introduced by Yust [35] (see Figure 16).[9]

Less obviously, traditional tonal-harmonic analysis of music is also hierarchical: spans of music are identified as being in a key, which are identified as subordinate to longer spans which might be goverened by a different, related, key. This is most evident in the "keyscale plots" presented by Sapp [30], whose triangular shape shows the structural similarities to the reduction trees of Lerdahl and Jackendoff (see Figure 4).

Not surprisingly, in the domain of digital music research, such as algorithmic analysis, classification, retrieval or composition, it is common to find that hierarchical representations of several kinds are used. Below, different examples are listed that illustrate the importance of this kind of representation. For automated composition, Högberg et al., in the system *Wind in the willows*, proposed the use of tree transducers [12]. The same hierarchical structure type was used by Jacquemard et al. [15] for elegantly representing and transforming rhythm notation (see Figure 5). With the objective of measuring the similarity of music, "metric tree" structures were introduced by Rizo [28] (see Figure 6) and extended with k-testables by Bernabeu et al. [1]. With the

[8] The relationships which a paradigmatic analysis shows, however, could be represented in the form of a graph, and so representation of paradigmatic analyses might be possible within the general framework we propose here. We leave development of this possibility to future work.

[9] Yust argues that a graph better represents elaborations which depend on the melodic interval from one note to another; for the same reasons a graph representation was proposed in [22] but subsequently abandoned as excessively complex in computational terms.

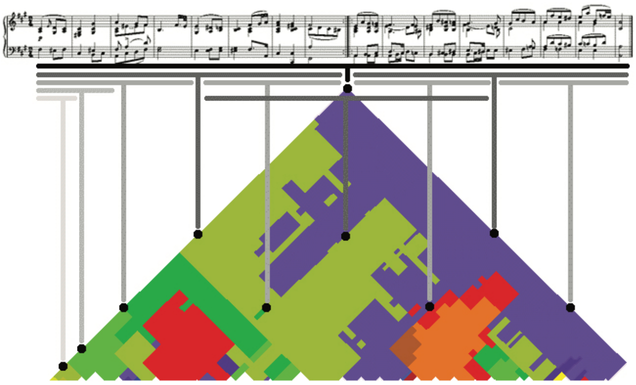


Fig. 4 Sapp's keyscape plot (extracted from [30], figure 3.14)

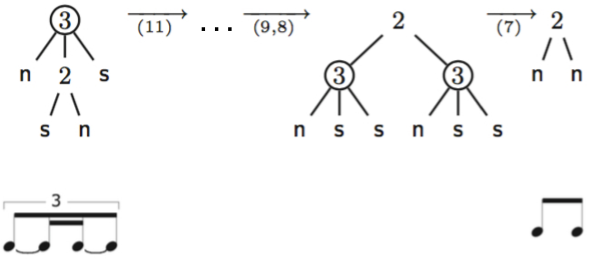
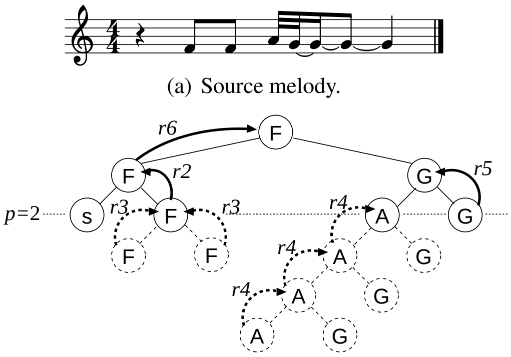


Fig. 5 Tree rewrite sequence proposed by Jacquemard et al. [15] (figure 7)



(a) Source melody.

(b) Propagation and prunning.

Fig. 6 Rizo's metric trees propagation and prunning [28] (figure 3.28).

same purpose, Pinto and Tagliolato [26] proposed a graph representation of melodic sequences (see Figure 7). Finally, several syntactic analysis systems have been proposed that produce parse trees such as the work by Gilbert and Conklin [5] for melody reduction, the contribution by Bod [2] for formalizing the perception of phrase structures (see Figure 8), or the syntactic perspective of simple rhythm structures by Lee [18].

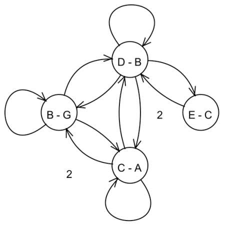


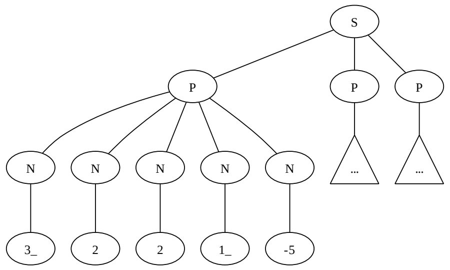
Fig. 7 Pinto's graph encoding of a sequence. Figure 6 in [26]

```
(3_221_-5)(-533221_-5)(13335432)(13335432_)(3_221_-5_)
```

(a) Bracket representation for folksong K0029, "Schlaf Kindlein feste"

```
S(P(N(3_)N(2)N(2)N(1_)N(-5))P(...)P(...) ... )
```

(b) Labeled bracket representation for (first five notes of) the structure above



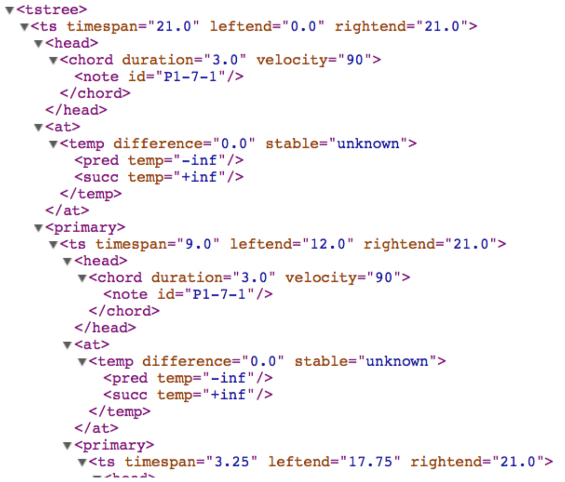
(c) Graphical representation of parenthesized notation above as found in [28] (figure 3.6)

Fig. 8 Example of data used to learn Bod's grammar in [2] (figures 7 and 8).

3 Existing hierarchical representations

Many of the approaches for hierarchical analysis and representations introduced above have points in common that could foster interesting synergies, but these are currently difficult to achieve due to the different representations and encodings of nevertheless similar concepts. A first humble attempt to normalize some representations was made by Rizo [28] (see figures 3.11 and 3.14 in that work) where standardized tree representations were offered to model different melodic reduction strategies such as the GTTM [19] and Marsden's elaboration trees [23]. However, no proposal for encoding datasets of this kind of tree structures was offered. Thus, the field of potentially useful standardization of many different *ad-hoc* encodings of similar musical-analysis and intermediate representation remains open.

To illustrate the issues, we discuss the case of the valuable SCHENKER41 database of analyses by Kirlin [16]. Each analyzed work is encoded in two files: the music itself is encoded in a MusicXML file [7], and the analysis is encoded in a text file with extension `.analysis` (see Figure 16). No grammar to read the format of these files is given, and in the paper [16] or the section 4.2 of Kirlin's PhD thesis [17] only the encoding of the notes based on scientific pitch and



```
▼<tstree>
  ▼<ts timespan="21.0" leftend="0.0" rightend="21.0">
    ▼<head>
      ▼<chord duration="3.0" velocity="90">
          <note id="P1-7-1"/>
        </chord>
      </head>
    ▼<at>
      ▼<temp difference="0.0" stable="unknown">
          <pred temp="-inf"/>
          <succ temp="+inf"/>
        </temp>
      </at>
    ▼<primary>
      ▼<ts timespan="9.0" leftend="12.0" rightend="21.0">
        ▼<head>
          ▼<chord duration="3.0" velocity="90">
              <note id="P1-7-1"/>
            </chord>
          </head>
        ▼<at>
          ▼<temp difference="0.0" stable="unknown">
              <pred temp="-inf"/>
              <succ temp="+inf"/>
            </temp>
          </at>
        ▼<primary>
          ▼<ts timespan="3.25" leftend="17.75" rightend="21.0">
```

Fig. 9 GTTM Time-span encoding extract Hamanaka et al. [9]

the prolongation operation are described. Once the developer opens the `.analysis` file, he/she has to figure out the meaning of some of the elements not described in the paper. In [17] Kirlin says that "The text file description is more relaxed than the MOP representation to allow for easy human creation of analyses." But SCHENKER41 is proposed as a corpus for use in other music analysis applications, which means that its format must also be readable by other computer software.

Another notable corpus has been produced by Hamanaka et al. [9] containing 300 musical pieces analyzed using the GTTM paradigm.[10] For encoding music, the MusicXML [7] format was used. The different GTTM analyses (grouping, metrical, time-span, prolongational, and harmonic) were encoded using several *ad-hoc* XML grammars (see Figure 9).

All these encodings share the same drawbacks. When music content is kept in a file different from that of the intermediate or analysis representation, tricks to unambiguously link analysis to music are devised that in many cases complicate the parsing of the format (see Figure 16). In cases where musical content and analysis is recorded in the same file, usually the original musical content is modified to accommodate the specifics of the representation (in Figure 6 long notes in the original score are replaced by tied shorter ones to be suitable for representation using the tree structure), and, even worse, only minimal content is kept, discarding what the representation does not need but could be useful in the future.

The main problem that arises from having so many different encodings comes from the fact that in order to represent similar concepts different parsers have to be built with nuances that are not always cross-compatible. In the following sections a standard encoding proposal is offered that tries to overcome these problems.

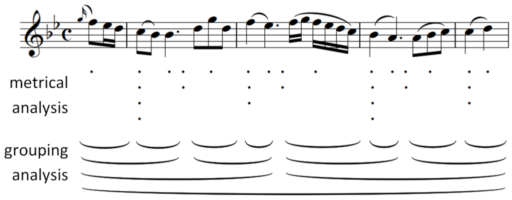


Fig. 10 Metrical and grouping analysis in GTTM style

4 Requirements of the encoding

Here we identify the requirements of a standard for the representation of hierarchical music analyses, taking into account the observations of the previous sections. The encoding must allow linked data requirements as detailed in [34]. Analytical and musical information need to be associated, and to facilitate this it should be possible to record the analysis and music in the same file. Section 2 has introduced the wide variety of kinds of analysis in use. It would be presumptuous to try to cover all of these in our proposals, and in any case it is important to be open to kinds of analyses which might emerge in the future. We therefore require that the standard allow for extensions with minimal effort.

Many segmentation analyses and elaboration reductions are usually represented using nested slurs (see Figure 10). As those slurs can be represented hierarchically, tree structures will be used for encoding this kind of analysis.

GTTM explicitly prohibits the crossing of branches and avoids gaps between them. Nevertheless, we do not want to exclude the possibility of future analyses which contain crossing branches or gaps, even if these are used only to illustrate the problems of such an analysis. We therefore do not make it a requirement of the standard that these kinds of branching are prohibited.

Apart from the hierarchical structure itself, one of the most important elements in the structures is the node, which can be a terminal or intermediate element of the tree or graph. It encodes most of the semantics of an analysis. The meaning of a node varies, depending on the approach. Some examples are Marsden's elaborations in Figure 3, the *left* and *right* branching in GTTM that denote which note is more important, or the music content embedded in Rizo's metric trees (see Figure 6) or Pinto's graphs (Figure 7) and Yust's MOP (Figure 16). Sometimes the original graphical representations have to be interpreted as being node information. In GTTM, for example, cadences[11] that are represented by using ellipse shapes that connect branches can be encoded as node labels. In any case, nodes must be able to contain both analysis information and links to the musical events referred to.

[10] `http://gttm.jp/gttm/`

[11] A cadence is sequence of at least two chords helping to conclude a musical fragment.

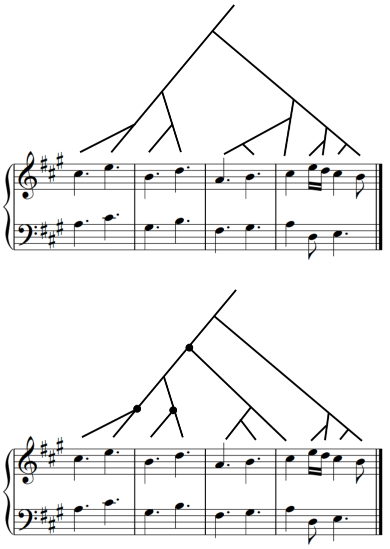


Fig. 11 Two different types of reductions of the same musical segment

Another essential feature of any analysis encoding is the ability to represent simultaneous alternative analyses of the same musical event or segment. Figure 11 shows different reduction approaches[12] to the same extract of music. In addition to concurrent analysis, annotations must be allowed to record alternative explanations, whether originating from an analyst or generated by a computer algorithm. It is not uncommon for analysts to disagree about the interpretation of a particular passage, or even for a single analyst to recognize that there are alternative interpretations. A large quantity of the literature about music analysis is concerned with the discussion of such cases, so it is essential to be able to represent alternative analyses in a way which allows accurate comparison.

Sometimes not everything in a hierarchical tree-like representation can be strictly encoded in a tree. In some cases links are made between the branches of a tree, as proposed in [23] and implemented in [24]. The "retained cadence" concept of Lerdahl and Jackendoff (indicated in a tree by the use of an ellipse shape between two branches) can also be considered as a case of such linking. In these cases, the structure is no longer strictly a tree, but instead a graph. Formally, the set of possible trees is a subset of the set of possible graphs, so any system for representing graphs is also capable of representing trees. From this perspective, it is not necessary to have a specific mechanism for representing trees but we have chosen to retain the tree type explicitly because trees have properties not found in all graphs which can facilitate efficient processing. (An example is simple branch pruning operations (see Figures 2 and 6) which are simple to perform on a tree but complex if done on a graph.) We believe that the benefit of this will outweigh the complication of dealing with the rare special cases of non-tree elements in an otherwise tree-like representation.

On the other hand, some analyses and intermediate representations require full graph encodings as previously indicated (Figures 7 and 16). These could be encoded as trees by unfolding the graphs and using the mechanism for representing linking between branches as mentioned above, but the semantics of the graph itself would be lost. We therefore consider it important to be able to use a graph representation when appropriate.

Finally, following the principle of encoding music and analysis together, it should be possible to record the results of analytical processes such as reductions in the same file as well. In many cases, notations such as GTTM trees implicitly encode the reduction in the tree. For example, a left branching implies that the content represented by the subtree on the right branch is more important than the content on the left branch, so the reduction can be accomplished just by pruning the left branch and retaining the notes in the right branch. However, there are situations where those reductions need to be explicitly encoded because they are not implied by the tree structure. In the case of a "fusion" branching in GTTM, for example, some notes are retained from both the left and right branch, and the representation needs to encode exactly which. In the case of a Roman-numeral analysis, the numeral and perhaps other information will need to be encoded to represent the interpretation of the harmony in this span (see Figure 17). In other cases (e.g., in a ground-truth corpus for machine-learning purposes), it is not necessary but more useful to represent the reduction results at the node concerned.

5 Music encoding initiative customization

Several formats have been proposed to represent music in a standardized way [32]. We are aware of one other proposal for the standardization of analytical processes [13] but it has a different scope from our project and lacks the embedding of music and analysis together. The XML format named Music Encoding Initiative (MEI) [10], based on the successful experience of the Text Encoding Initiative (TEI) [14], is able to encode music of different notations, such as mensural and common western notation, together with metadata for bibliographic cataloguing. We have chosen to use this representation as the basis for our proposal because of its wide musical scope, its sound design, and its strong support. Furthermore, it natively allows the use of linked data technologies over the encoding itself.

MEI has an analysis module available. It has two parts, the first one focused on relating elements inside the musical score, the second one allowing the encoding of event-based analytical information such as melodic and pitch-class analysis. Being founded on an XML format, MEI is inherently

[12] A reduction is the removal of what a given analysis technique considers as non-essential or less important notes (see Figure 13).

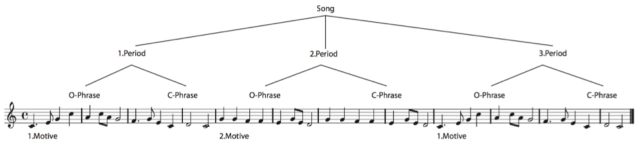


Fig. 12 Form analysis from [20].

hierarchical. If the hierarchical analysis matches the beginning and ending of work divisions, voices or measures (see Figure 12), the XML structure itself could be used to encode analyses. However, this is not always the case (see Figure 1). It is therefore necessary to make use of some other mechanism to explicitly represent the hierarchy of the analysis and not to rely on the inherent hierarchy of the XML analysis of the music for this purpose.

The MEI initiative is conceived to accommodate any kind of musical content, for which it has been equipped with an extension and customization mechanism [10]. This device enables the incorporation of our needs, taking advantage of all MEI infrastructures. In particular, we have reused all musical content elements from MEI and some of its analysis module that will be used to encode the musical content of nodes and the results of analytical processes such as reductions (e.g., the bottom system in Figure 13). MEI does not have components for explicitly encoding tree or graph structures but its parent format, TEI, includes such structures in the *net* module that we will adapt to fulfil our needs.

5.1 Semantic encoding

We stated above that one of the objectives of a representation should be to preserve the knowledge embodied in an analysis. This implies that the representation should be, to some degree, "semantic." Ideally it should be possible to determine unambiguously what the representation "means," and analyses with different meanings should have different representations. This means that the elements of the analysis should not be simply those elements which represent the tree or graph structure (nodes and arcs) but that they should also make explicit the nature of the entity represented by a node or the relationship represented by an arc. In fact, this is a design principle of MEI.

However, this ideal is neither always possible nor desirable. For one thing, the underlying analysis can be ambiguous, and a degree of interpretation might be required from the person reading the analysis. The encoding might overcome this by representing a particular interpretation unambiguously. We have mentioned above that it is necessary to have a means of representing alternative analyses, and the same mechanism could represent alternative interpretations. As an example, consider the notation which Schenker sometimes uses where a slur turns back on itself in a sigmoid shape. This occurs twice in Figure 3 *(a)*. In the bass (the lower of the two staves of music) it occurs in conjunction with a normal, smaller, slur from the fourth to the fifth note of the scale, and indicates a common progression[13] which Schenker always notates in this fashion (this is explained in paragraph 56 of *Der Freie Satz* [31, p. 30]). In the melody (the upper staff), it is not clear whether the turn in the slur is simply to avoid a collision with some text above (not shown in the figure) or whether it is intended to show something meaningful about the melodic progression here (perhaps the repetition of the notes D and C sharp). Later in the same figure in the original, a parallel passage has a similar turning slur in the melodic line but there is no text with which it might collide. On the other hand, in another graph earlier in *Der Freie Satz* [31, fig. 87, no. 5], Schenker analyzes the same melodic passage in a similar way but without slurs which turn. The evidence for whether or not this notation in the melodic line is meaningful or accidental is therefore contradictory.

Furthermore, it is sometimes desirable to process analyses which have different origins and which will therefore, in a fully semantic encoding, be represented according to different semantic frameworks. For example, we might want to compare analyses in the GTTM format with Schenkerian analyses. Some of the detail will be incomparable, but the basic tree structures will be comparable and might, for example, indicate whether or not a passage is considered to be segmented in the same way. We therefore propose mechanisms which allows some representations to be semantic and others to be non-semantic (representing only the structure of the essential tree or graph).

5.2 Encoding examples

All requirements expressed in the section 4 have been adopted into our proposal. The result is a self-contained music and analysis format that allows both a semantic and a non-semantic encoding. As we cannot give here a comprehensive explanation of all its elements, only some representative examples are offered.

As has been mentioned above, in some cases a semantic encoding with explicit elements only applicable to the analysis type at issue is desirable over a more general non-semantic encoding that explicitly reflects the tree or graph structure. For each analysis format, we propose two alternative encodings: a semantic schema and a non-semantic one. An XSLT for converting from the semantic XML to the non-semantic structure can be easily prepared, in some cases the reverse transformation may not be possible because of ambiguities of interpretation such as illustrated in the discussion of turning slurs above.

[13] A progression is a sequence of chords or harmonic functions.

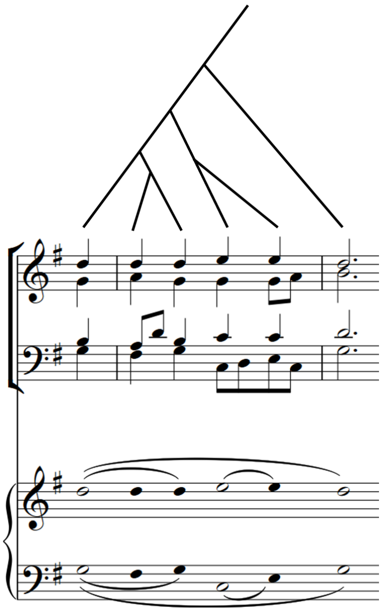


Fig. 13 The bottom part contains the result of the GTTM reduction depicted in the tree above

It will be the encoding author who determines which encoding approach is more suitable for the objective of the encoding.

In this section three different types of semantic encodings of analyses are detailed and their non-semantic equivalent encodings are presented. In order to express in a compact way a Relax NG Compact Syntax[14] is used rather than showing the complete ODD[15] file as used in TEI [14] and MEI [10].

5.3 GTTM basic tree encoding

As mentioned above, we use elements from MEI's parent, TEI, for the representation of trees and graphs. Listing 1 shows an extract of the first, incomplete measure of the GTTM reduction in Figure 13. Only essential elements for understanding the example have been kept in the XML. Both music and analysis are encoded in the same file: the music extract is shown in the listing inside an MEI `<measure>` element; the analysis is encoded using an `eTree` element, adapted from the *net* module of TEI. Branches of the tree are also encoded using `eTree` elements (since by definition branches of a tree are also trees). The leaves are represented using `eLeaf` elements, which formally are nodes as well. Each node contains a `label` element which encodes additional useful information. In the case of the leaf, for example, the alignment with the music analyzed is encoded.

```
<mei>
    <!-- ... removed to save space ... -->
    <measure xml:id="meas0" type="anacrusis">
        <staff n="1">
            <layer n="1">
                <note xml:id="n1" dur="4" oct="5"
                     pname="d"/>
            </layer>
            <layer n="2">
                <note xml:id="n6" dur="4" oct="4"
                     pname="g"/>
            </layer>
        </staff>
        <staff n="2">
            <layer n="3">
                <note xml:id="n13" dur="4" oct="3"
                     pname="b"/>
            </layer>
            <layer n="4">
                <note xml:id="n20" dur="4" oct="3"
                     pname="g"/>
            </layer>
        </staff>
    </measure>
    <!-- ... removed to save space ... -->
    <eTree type="gttm-prolongational-reduction">
        <label type="right-branching" subtype="
             progression"/>
        <!-- ... removed to save space ... -->
        <eTree>
            <label type="right-branching" subtype="
                 progression"/>
            <eLeaf xml:id="timeSlice0">
                <label>
                    <timespan startid="#meas0"
                         tstamp="4" tstamp2="0m+5"/
                         >
                    <!-- not necessary, it could be
                         used just timespan -->
                    <note sameas="#n1"/>
                    <note sameas="#n6"/>
                    <note sameas="#n13"/>
                    <note sameas="#n20"/>
                    <summary>
                        <note sameas="#n1"/>
                        <note sameas="#n20"/>
                    </summary>
                </label>
            </eLeaf>
            <!-- ... removed to save space ... -->
```

Listing 1 Extract of encoding of analysis in Figure 13 (first, incomplete measure)

The leaf represents a time slice which in this case spans the measure. This time span is represented using a `timespan` element introduced by us that contains attributes for setting the measure the segment starts (`startid`) and metrical positions to show the beginning (`tstamp`) and ending (`tstamp2`) of the time span following the MEI guidelines. Some other combinations of MEI attributes that unambiguously identify timespans or individual events have been used in other encodings. In addition, this label contains a link to the notes in the musical part of the encoding using the `sameas` attribute from the analysis module and `xml:id` attributes that anchor the target notes. The other parts of the encoding add information about the analysis. The `type` attribute of the root node shows that this analysis contains a GTTM prolongation reduction, and other `label` elements with `type` attributes show the kind of *elaboration* the nodes

[14] `http://relaxng.org/compact-tutorial.html`

[15] 'One Document Does it all', a literate programming language for XML schemas.

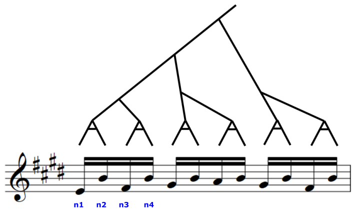


Fig. 14 Time-span reduction with "'fusion" branching (See [19], fig. 7.11)

represent. Finally, the `summary` element inside the leaf label allows the product of the analysis, in this case a reduction or music summarization, to be emitted.

The content of these `label` elements and `type` attributes are semantic, in the sense used above, but they are attached to a representation which is fundamentally not semantic beyond the basic tree structure which it conveys. A system using this representation would not be able to make assumptions about the presence of these elements and attributes, for example. In the following sections we make proposals for semantic encoding schemes, with examples of encodings using both these schemes and, for the reasons stated above, equivalent representations in a non-semantic scheme.

5.3.1 GTTM Time-span reduction

Listing 2 shows an example of our proposal for the semantic encoding of GTTM time-span reduction.

The `letter` attribute is used for labelling levels [19, p. 132]. The different kinds of branching (which have different meanings [19, p. 128]), are encoded using the `branching` attribute whose values are described as follows: let *x* be the left tree and *y* the right tree, "right" value is used when *x* dominates *y* and "left" value when *y* dominates *x*. In the case of ternary branching, with subtrees *x*, *y*, *z*, "left" means *z* dominates *x* and *y*, "right" means *x* dominates *y* and *z*, and "center" is used when *y* dominates both *x* and *z*. GTTM proposes two other branching kinds, "fusion" as shown in Figure 14, and "cadentialRetention". In these cases, the contents of the reduction are not simply the contents of one child or the other but a combination of their contents.

This format has been used to encode the music excerpt in Figure 14 into the XML shown in Listing 2. Note that only the analysis part of the XML is shown. An equivalent non-semantic encoding is presented in Listing 3.

5.3.2 GTTM Prolongational reduction

This is the kind of reduction introduced previously in the non-semantic encoding of example of Listing 1. The grammar in Listing 4 is a proposal for the semantic encoding of GTTM prolongational reduction. This kind of tree usually contains two children, encoded using two successive `prolongationalReduction` elements. The `branching` attribute is similar to that used in the time-span reduction, but with different meaning. In this case, "right" denotes tensing motion and "left" denotes relaxing motion (see page 185 in [19]).

```
<timeSpanReduction branching="right">
    <!-- removed to save space -->
    <timeSpanReduction branching="right">
        <timeSpanReduction branching="fusion">
            <events>
                <note sameas="#n1"/>
            </events>
            <events>
                <note sameas="#n2"/>
            </events>
        </timeSpanReduction>
        <timeSpanReduction branching="fusion">
            <events>
                <note sameas="#n3"/>
            </events>
            <events>
                <note sameas="#n4"/>
            </events>
        </timeSpanReduction>
    </timeSpanReduction>
    <!-- removed to save space -->
</timeSpanReduction>
```

Listing 2 Time-span reduction using a semantic encoding

```
<eTree type="gttm-time-span-reduction">
    <!-- Removed to save space -->
    <eTree>
        <label type="right-branching"/>
        <eTree>
            <label type="fusion-branching"/>
            <eLeaf>
                <label>
                    <note sameas="#n1"/>
                </label>
            </eLeaf>
            <eLeaf>
                <label>
                    <note sameas="#n2"/>
                </label>
            </eLeaf>
        </eTree>
        <eTree>
            <label type="fusion-branching"/>
            <eLeaf>
                <label>
                    <note sameas="#n3"/>
                </label>
            </eLeaf>
            <eLeaf>
                <label>
                    <note sameas="#n4"/>
                </label>
            </eLeaf>
        </eTree>
    </eTree>
    <!-- Removed to save space -->
</eTree>
```

Listing 3 Time-span reduction using a non-semantic encoding

An example of the analysis part of the semantic encoding of Figure 15 is shown in Listing 5, and an equivalent non-semantic encoding is given in Listing 6.

```
element prolongationalReduction {
        attribute branching {”left” | ”right”},
    attribute connectionKind {”progression” |
        ”weak” | ”strong” },
    element {prolongationalReduction},
    element {prolongationalReduction},
    element reduction {
        element reductionNote {
                attribute copyOf {xml:id}
        }+
    }?
    |
    element events {
        element timespan {
                 attribute startid {xml:id},
             attribute tstamp {xml:id},
             attribute tstamp2 {xml:id},
             attribute type {”regular” | ”augmented”}
        }
        |
        element note {
                attribute sameas {xml:id}
        }+
    }
}
```

Listing 4 GTTM Prolongational reduction semantic encoding

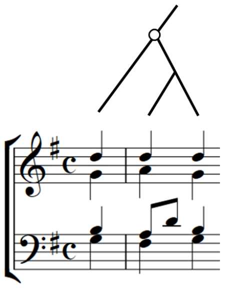


Fig. 15 Time-span reduction example with a “progression” and a “‘strong” branching

```
<prolongationalReduction branching=”right”
     connectionKind=”strong”>
    <events>
        <timespan startid=”#meas0” tstamp=”4”
             tstamp2=”1m+1”/>
    </events>
    <prolongationalReduction branching=”left”
         connectionKind=”progression”>
        <events>
            <timespan startid=”#meas1” tstamp=”1”
                 tstamp2=”0m+2”/>
        </events>
        <events>
            <timespan startid=”#meas1” tstamp=”2”
                 tstamp2=”0m+3”/>
        </events>
    </prolongationalReduction>
</prolongationalReduction>
```

Listing 5 Prolongational reduction using a semantic encoding (see Fig. 15)

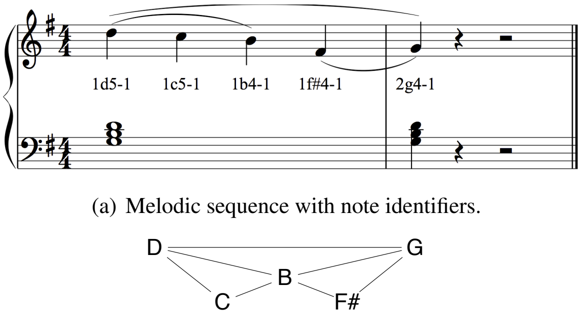
```
<eTree type=”gttm−prolongational−reduction”>
    <label type=”right−braching” subtype=”strong”/>
    <eLeaf>
        <label>
            <timespan startid=”#meas0” tstamp=”4”
                 tstamp2=”1m+1”/>
        </label>
    </eLeaf>

    <eTree>
        <label type=”left−braching” subtype=”
             progression”/>
        <eLeaf>
            <label>
                <timespan startid=”#meas1” tstamp=”
                    1” tstamp2=”0m+2”/>
            </label>
        </eLeaf>
        <eLeaf>
            <label>
                <timespan startid=”#meas1” tstamp=”
                    2” tstamp2=”0m+3”/>
            </label>
        </eLeaf>
    </eTree>
</eTree>
```

Listing 6 Prolongational reduction using a non-semantic encoding (see Fig. 15)

5.3.3 MOP graph encoding

Another example, this time encoding a graph (from Figure 16), is shown in Listing 7. The same principles hold for encoding the music content. Now the structural elements of the analysis, all of them adapted from the *net* module of the TEI format, are the `graph`, `node`, and `arc` elements. The `node` encoding follows the same principles as the `eLeaf` in the case of trees. The `arc` represent the edges between nodes and contain the semantics of the edge, in this case the type of MOP elaboration, in the `label` element.



(a) Melodic sequence with note identifiers.

(b) MOP representation of this music.

```
1d5-1 (1c5-1) 1b4-1
(1f#4-1) 2g4-1
1d5-1 (1b4-1) 2g4-1
```

(c) An encoding of this MOP representation.

Fig. 16 MOP encoding as found in SCHENKER41 dataset (figures extracted from [16], figures 1, 3, and 2).

```
<mei>
    <!-- ... removed to save space ... -->
    <measure xml:id="meas1" n="1">
        <staff n="1">
            <layer n="1">
                <note xml:id="n1" dur="4" layer="1"
                    oct="5" pname="d"/>
                <note xml:id="n2" dur="4" layer="1"
                    oct="5" pname="c"/>
                <note xml:id="n3" dur="4" layer="1"
                    oct="4" pname="b"/>
                <note xml:id="n4" dur="4" layer="1"
                    oct="4" pname="f" accid="s"/>
            </layer>
        </staff>
        <!-- ... removed to save space ... -->
    </measure>
    <!-- ... removed to save space ... -->
    <graph type="MOP">
        <node xml:id="vD"> <!-- v = vertex -->
            <label>
                <note sameas="#n1"/>
            </label>
        </node>
        <node xml:id="vC">
            <label>
                <note sameas="#n2"/>
            </label>
        </node>
        <node xml:id="vB">
            <label>
                <note sameas="#n3"/>
            </label>
        </node>
        <arc from="vD" to="vC">
            <label type="elaboration" subtype="
                elaboratedNode" value="vC"/>
        </arc>
        <arc from="vB" to="vC">
            <label type="elaboration" subtype="
                elaboratedNode" value="vC"/>
        </arc>
        <!-- ... removed to save space ... -->
```

Listing 7 Extract of encoding of Figure 16 (first three notes)

MOP structures can be semantically encoded following the grammar in Listing 8. Using it, the example shown above in Figure 16 that was non-semantically encoded in Listing 7, is codified semantically as shown in Listing 9 (only the analysis content is shown to save space).

5.3.4 Roman-numeral analysis

A tonal analysis of a work decomposes a work into tonal regions with a key and in turn divides each one into chords with some tonal function assigned, establishing the melodic function of each note in the context of the underlying harmony. While this kind of analysis is not usually represented in a hierarchical manner, it can, as in many other cases, be expressed as a tree-like structure. Figure 17(a) shows an example. The melodic analysis can be encoded using the `@mfunc` attribute from the MEI format. The tree perfectly reflects the nature of the key, tonal function, degree hierarchy. The only case in which the resulting structure is not strictly a tree can be found in the pivot chords that belong at the same time to the two different branches encoding a modulation,[16] the first branch representing the source key and the second branch defining the target key. In this case, the tree leaf that encodes the pivot chord has two parents, the left one belongs to the hierarchy of the source key and the right parent that corresponds to the hierarchy of the target key (see Figure 17(b)).

This structure can be encoded both semantically and non-semantically using the same approach used in the examples of GTTM and MOP introduced above. The encoding of pivot chords is performed as a strict tree where the labels of both leaves point to the same event, the one containing the pivot chord.

```
element mop {
        element elaboration {
        attribute from {xml:id},
        attribute to {xml:id},
        attribute prolongation {xml:id}
    }+
}
```

Listing 8 MOP semantic encoding. `xml:id` is a MEI attribute that references a MEI element

```
<mop>
    <elaboration from="#nD" to="#nB" prolongation="
        #nC"/>
    <elaboration from="#nB" to="vG" prolongation="#
        nF"/>
    <elaboration from="#nD" to="#nG" prolongation="
        #nB"/>
</mop>
```

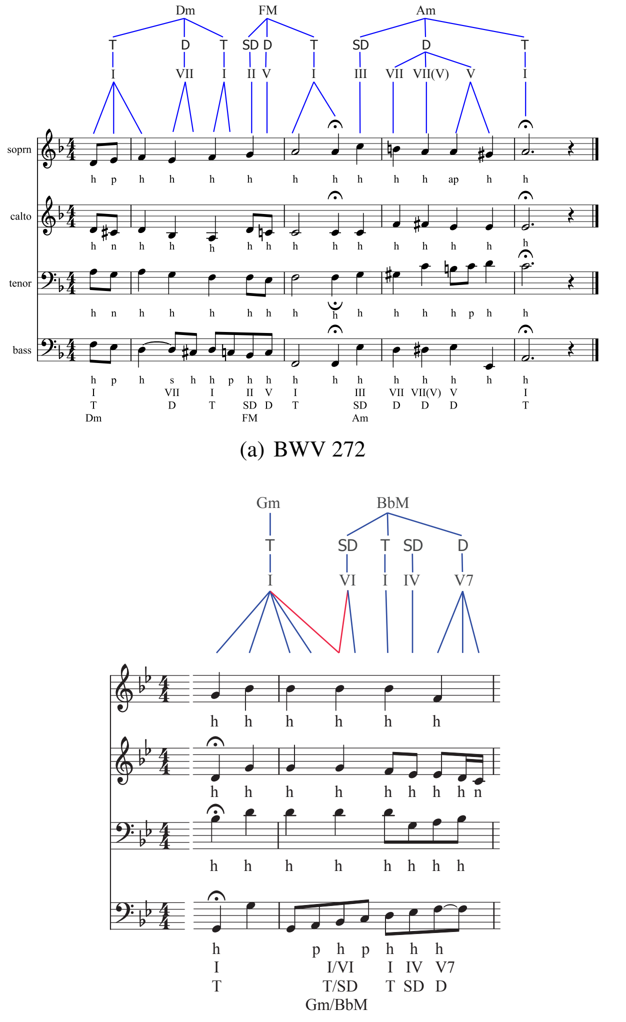
Listing 9 Semantic encoding of Figure 16

6 Conclusions

If format standards did not exist digital libraries would not be possible, and all possible research based on data and its further application to methods useful for society. This condition, while obvious, is not always met. There are formats for encoding music content, either notated or played, metadata describing content or the source of music documents, and some standards allowing the encoding of critical apparatus. However, few standards exists yet for the encoding of the analysis of musical works. It seems that in the near future, methods in digital musicology will not only require standard formats to encode music and metadata, but also standard representations of the output of analytical processes, both as a means for sharing results and as an intermediate step for building more complex systems that have as input the yield of music analysis methods.

In this work we have introduced a proposal for a standard for the encoding of hierarchical analyses and intermediate representations of music that fulfils these needs.

[16] A modulation is a temporary change of key.



(a) BWV 272

(b) BWV 274

Fig. 17 The Roman-numeral analysis below the system can be encoded in the form of a tree. Extracts from two Bach chorales.

Our proposal has recently been presented at the Music Encoding Initiative Conference 2016 in Montréal and Digital Libraries for Musicology 2016. This paper substantially extends the text in the proceedings of that conference [29], including more examples and introducing semantic encodings. Our encoding proposals are under continual development and are being steadily improved and the final elements and attributes may slightly change in the near future. Nevertheless, the core idea and principles are expected to be maintained.

Complete details, examples, and the customization files and grammars can be found online and updated at `http://grfia.dlsi.ua.es/cm/worklines/mei-ha`.

Acknowledgements We would like to thank Eleanor Selfridge-Field, Don Byrd, Tom Collins, and Phillip Kirlin for contributions during the MEC 2016 and DLfM 2016 conferences which have informed this work, and to thank Ichiro Fujinaga and Perry Roland for their encouragement to continue it.

References

1. José F. Bernabeu, Jorge Calera-Rubio, José M. Iñesta, and David Rizo. Melodic Identification Using Probabilistic Tree Automata. *Journal of New Music Research*, 40(2):93–103, June 2011.
2. Rens Bod. A General Parsing Model for Music and Language. *Music and Artificial Intelligence*, pages 77–90, 2002.
3. John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference, Miami, Florida, USA, 2011*, pages 633–38, 2011.
4. György Fazekas, Yves Raimond, Kurt Jacobson, and Mark Sandler. An overview of semantic web activities in the omras2 project. *Journal of New Music Research*, 39(4):295–311, 2010. doi: 10.1080/09298215.2010.536555.
5. Édouard Gilbert and Darrell Conklin. A probabilistic context-free grammar for melodic reduction. *International Workshop on Artificial Intelligence and Music at IJCAI-07. Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India*, 2007.
6. Robert O. Gjerdingen. *A Classic Turn of Phrase: Music and the Psychology of Convention*. University of Pennsylvania Press, Philadelphia, 1988.
7. Michael Good and Geri Actor. Using MusicXML for file interchange. *Web Delivering of Music, International Conference on*, 0:153, 2003. doi: http://doi.ieeecomputersociety.org/10.1109/WDM.2003.1233890.
8. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Implementing “A Generative Theory of Tonal Music”. *Journal of New Music Research*, 35(4):249–277, December 2006.
9. Masatoshi Hamanaka, Keiji Hirata, and Satoshi Tojo. Musical Structural Analysis Database Based on GTTM. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 325–330, 2014.
10. Andrew Hankinson, Perry Roland, and Ichiro Fujinaga. The Music Encoding Initiative as a Document-Encoding Framework. *ISMIR*, 2011.
11. Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music*

Information Retrieval, London, UK, 2005, pages 66–71, 2005.

12. Johanna Högberg. Wind in the Willows - Generating Music by Means of Tree Transducers. In *CIAA*, pages 153–162, 2005.
13. Eric J. Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M. Bittner, and Juan Pablo Bello. JAMS: A JSON Annotated Music Specification for Reproducible MIR Research. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 591–596, 2014.
14. Nancy M. Ide and C. M. Sperberg-McQueen. The TEI: History, goals, and future. *Computers and the Humanities*, 29(1):5–15, 1995.
15. Florent Jacquemard, Pierre Donat-Bouillud, and Jean Bresson. A Structural Theory of Rhythm Notation Based on Tree Representations and Term Rewriting. In *Advances in Artificial Intelligence*, pages 3–15. Springer International Publishing, Cham, June 2015.
16. Phillip B. Kirlin. A data set for computational studies of Schenkerian analysis. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27-31, 2014*, pages 213–218, 2014.
17. Phillip B. Kirlin. *A probabilistic model of hierarchical music analysis*. PhD thesis, University of Massachusetts Amhers, Massachusetts, 2014.
18. C. S. Lee. The rhythmic interpretation of simple musical sequences: Towards a perceptual model. In R. West, P. Howell, and I. Cross, editors, *Musical Structure and Cognition*, pages 53–69. Academic Press, London, 1985.
19. Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.
20. Christiane Linster. On analyzing and representing musical rhythm. In Mira Balaban, Kemal Ebcioğlu, and Otto Laske, editors, *Understanding music with AI: perspectives on music cognition*, pages 414–427. AAAI Press/MIT Press, Menlo Park, California, USA, 1992.
21. Pasquale Lisena, Konstantin Todorov, Cécile Cecconi, Françoise Leresche, Isabelle Canno, Frédéric Puyrenier, Martine Voisin, Thierry Le Meur, and Raphaël Troncy. Controlled vocabularies for music metadata. In *Proceedings of the 19th ISMIR Conference, Paris, France, September 23-27, 2018*, pages 424–430, 2018.
22. Alan Marsden. Representing Melodic Patterns as Networks of Elaborations. *Computers and the Humanities*, 35(1):37–54, 2001.
23. Alan Marsden. Generative structural representation of tonal music. *Journal of New Music Research*, 34(4): 409–428, December 2005.
24. Alan Marsden. Schenkerian Analysis by Computer: A Proof of Concept. *Journal of New Music Research*, 39 (3):269–289, September 2010.
25. Jean-Jacques Nattiez. *Fondements d'une sémiologie de la musique*. Union Générale d'Éditions, 1975.
26. Alberto Pinto and Paolo Tagliolato. A generalized graph-spectral approach to melodic modeling and retrieval. In *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*, pages 89–96, New York, NY, USA, 2008. ACM.
27. Sabbir Rashid, Deborah L. McGuinness, and David De Roure. A Music Theory Ontology. In *International Workshop on Semantic Applications for Audio and Music - ISWC*, Monterey, Califormia, 2018.
28. David Rizo. *Symbolic music comparison with tree data structures*. PhD thesis, Universidad de Alicante, 2010.
29. David Rizo and Alan Marsden. A standard format proposal for hierarchical analyses and representations. In *Proceedings of the 3rd International workshop on Digital Libraries for Musicology*, pages 25–32, New York, New York, USA, 2016. ACM Press.
30. Craig Stuart Sapp. *Computational methods for the analysis of musical structure*. PhD thesis, Stanford University, 2011.
31. Heinrich Schenker. *Free Composition (Der Freie Satz)*. Longman, 1979 (1935).
32. Eleanor Selfridge-Field. *Beyond MIDI: The handbook of musical codes*. Beyond MIDI: The handbook of musical codes. MIT Press, Cambridge, Massachusetts, USA, 1997.
33. Giorgio Uboldi and Giorgio Caviglia. Information Visualizations and Interfaces in the Humanities. In *New Challenges for Data Design*, pages 207–218. Springer London, London, December 2014.
34. David M. Weigl, David Lewis, Tim Crawford, Ian Knopke, and Kevin R. Page. On providing semantic alignment and unified access to music library metadata. *International Journal on Digital Libraries*, Aug 2017. ISSN 1432–1300. doi: 10.1007/s00799-017-0223-9. URL `https://doi.org/10.1007/s00799-017-0223-9`.
35. Jason Yust. The Geometry of Melodic, Harmonic, and Metrical Hierarchy. In *Mathematics and Computation in Music*, pages 180–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.