Representing atypical music notation practices: An example with late 17th century music

Rodolfo Zitellini
Swiss RISM Office
rodolfo.zitellini@rism-ch.org

Laurent Pugin
Swiss RISM Office
laurent.pugin@rism-ch.org

ABSTRACT

From the 17th century to the first decades of the 18th century music notation slowly loses all its mensural influences, becoming virtually identical to what we would consider common modern notation. During these five decades of transformation composers did not just suddenly abandon older notation styles, but they used them alongside ones that would eventually become the standard. Void notation, black notation and uncommon tempi were all mixed together. The scholar preparing modern editions of this music is normally forced to normalise all these atypical notations as many software applications do not support them natively. This paper demonstrates the flexibility of the coding scheme proposed by the Music Encoding Initiative (MEI), and of Verovio, a visualisation library designed for it. The modular approach of these tools means that particular notation systems can be added easily while maintaining compatibility with other encoded notations.

1. INTRODUCTION

Mid to late 17th century musical notation was already very similar to what we know and use nowadays. The conventions began to be the same and many vestiges from the past were being lost. The Late Baroque period should not be considered as uniform, but as one of great transformation. This transition affected musical notation, which retained some specific features and idiosyncrasies.
Many particular features found towards the end of the century, such as void notation and coloration (see Sections 2.1 and 2.2), are sometimes considered to be just left over from the past with little practical use.
Nevertheless composers and printers used these alternative notation types extensively, making it important for modern editions not to lose them.

Up to the second half of the twentieth century, the most common custom for critical editions has been to transcribe the notation to the modern standard. Nowadays, on the other hand, more and more editions try to reproduce faithfully the features of the original sources [1]. Indeed, even if features such as void notation seem at first glance just a curiosity, completely removing them in transcriptions makes the edition less useful for philological work and scholars frequently have to reach out for difficult to find sources just to clarify some small passages.

There is a problem with the symbolic digital representation of music written in these non-standard notations, as currently no system is capable of encoding them properly. Since no notation system directly supports void or black notation (hand adjustments are necessary in many cases), the resulting encoded data is often invalid since, as we will see, it is often necessary to use over- or underfilled measures to represent the music.

In this paper, we investigate the use of the Music Encoding Initiative (MEI) scheme together with the rendering library, Verovio, for better encoding and visualising unusual notation features of the late 17th century. The following section explains more precisely some of these uncommon features we find in the music notation of that time, and then we look at previous work for both encoding and visualising them. Our proposed approach with MEI and Verovio is then explained and illustrated.

2. LATE 17TH CENTURY MUSIC

2.1 Black notation and colorations

Black notation is a feature of late 17th century music reminiscent of its mensural past. As in the mensural system it indicates particular rhythmic values, and its use is generally limited to simple cases. It is often mixed with white void notation, requiring, in many cases, great manual intervention to be typeset with modern systems.

Black notation is similar but not identical to modern notation. It was used in triple meter, 3/1 or 3/2, to indicate rhythmic alterations. The corpus in which such notation is found is vast and varied, and composers went to great pains to specify such notational details. It is therefore important that such information not be lost in modern editions, and such editions should encode it properly.

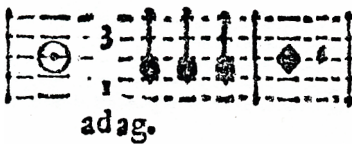

Figure 1. Black notation found in a sonata by Giovanni Maria Bononcini [2], bars 34-35 (excerpt from *Violino Primo* partbook).

Figure 1 shows a typical example of black notation in a late 17th century source. Typesetting it in software such as Finale requires adjustment to each note head or encoding it with an approximate symbolic representation. It is also important to note that the default music font in Finale does not include a black semibreve note head, so one is forced to use the smaller note head for quarters and other notes with flags.

Figure 2. Full-score modern edition Figure 1, bars 34-35,[3].

Figure 2 shows a the same passage as Fig. 1 transcribed into modern notation. In the introduction to the edition the author laments the lack of easy support for 17th century notation in the (unspecified) typesetter she is using [3]. Coloration of single notes is encountered much more often and is used often in triple meter. It is a leftover expression from the old mensural system, and, as coloration indicated at the time, it specifies that a perfect note loses its perfection:

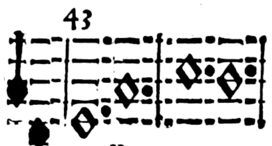

Figure 3. Bars 92-96 in a Mass by Maurizio Cazzati [4], excerpt from *Organo* partbook.

In the small extract shown in Fig. 3 the author employs coloration to visually underline a different rhythmic subdivision. This is an extremely common example, and modern editions have employed various strategies over the years to deal with it.

Normally brackets are used to signal coloration, but in some cases no indication at all is given in the score, only resorting to a small footnote in the critical apparatus.

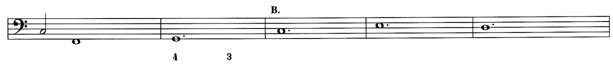

Figure 4. Basso of a full-score modern edition of Cazzati's *Messa Concertata* from op. 14, bars 92-95 [5].

As we can see from Fig. 4, although the modern notation used is correct and clear, it completely loses the visual cue given in the source. This kind of cue is not only useful to performers: they are also important to scholars particularly interested in notational problems. Burying them in lengthy critical notes makes this information inaccessible to the reader.

2.2 Void notation

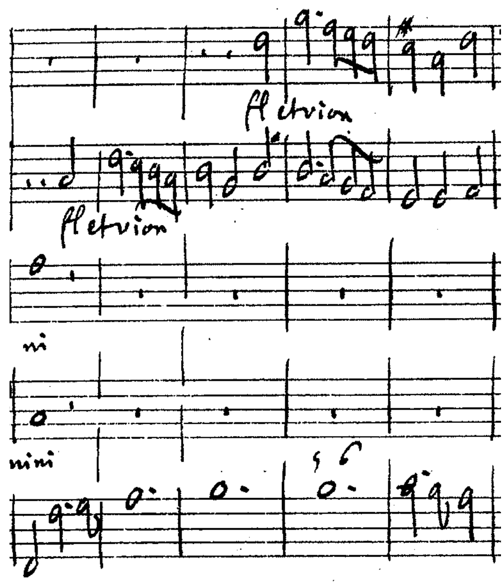

Figure 5. Marc-Antoine Charpentier, *In Honorem Sancti Xaverii canticum*. Facsimile of autograph ms., bars 110-114.

The use of void notation at the end of the 17th century is probably one of the great riddles still left for musicologists to figure out. Figure 5 gives us a common occurrence of this style of notation. In 3/2 time, quarter notes are written out as eighths with a white note head, and, if appropriate, beamed as "normal" eighths.

Various studies seem to show that it is simply an alternative to proper modern "black" notation, arbitrarily applied and without a specific meaning. Nevertheless it appears in copious prints and manuscripts [6]. Most well-known are the examples by Marc-Antoine Charpentier, but many others come from printed music in northern Italy, in the region of Bologna more specifically, a trend lasting until the mid-1710s [7].

Through a closer investigation of the issue it becomes apparent that void notation is not limited to a couple of sources here and there, but was a deliberate choice made by composers. Recent studies [8] have brought to light the fact that music printers were explicitly requesting movable type in void notation. This is the case, in the final years of the 17th century, of Silvani in Bologna, who purchased his characters in Venice. In his order [8], Silvani specifically laid out the details for void notation characters.

Void notation had commonly been used in the music printed in the City until the late 1650s, when Maurizio Cazzati, *Maestro di Cappella* in the Basilica of S. Petronio, produced the first examples of it [9]. Many later editions were produced in Bologna with this notation.

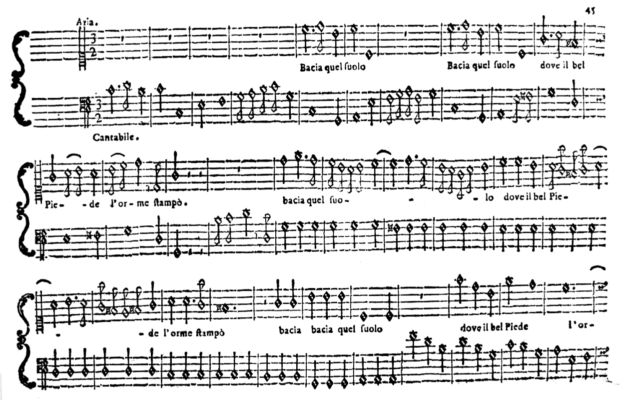

Figure 6. A late example of void notation in a collection of cantatas by Pirro Capicelli Albergati [10]. Many collections of music from this city are visually very similar to this setting [11].

Since the total corpus of music in void notation is somewhat restricted, scholars have mostly dismissed it in modern transcriptions. In recent years scholars wanting to maintain this original aspect of the sources have generally been discouraged by the lack of easy support for it in commercial music notation software.

Figure 7. Marc-Antoine Charpentier, *In Honorem Sancti Xaverii canticum*. Modern edition [12], bars 110-114.

In Figure 7 we see clearly how the void notation from Fig. 5 is reproduced using modern notation. A note from the editor at the beginning of the 3/2 section informs us that the original is in void notation, which is not maintained in the transcription. We cannot blame the editor, as a proper transcription would require manually editing each note to change the note head to a white one. Also, the original manuscript source clearly beams the white eighth notes. In this case it is not sufficient merely to change the note head, although, in the notation software, the beamed notes need to be set at actual eighth notes (to obtain the beam), resulting in an insufficient number of notes in the measure. Not only does this require a deal of manual adjustment but the resulting symbolic representation of the music is invalid.

3. PREVIOUS WORK

A plethora of music codes have been developed [13]. Mostly they have a specific application or notation type in mind. For common western music notation (CWMN) and music notation applications, the most widely used is MusicXML [14]. It is designed as an interchange format between computer music applications. Therefore, it is meant to be sufficient for most applications but not optimal. MusicXML has no support for early music notation (before 1600) and is not designed to be customizable. For earlier notations, one of the first solutions proposed was DARMS with a dialect for mensural notation [13]. More recently, an XML code was developed as part of the Computerized Mensural Music Editing (CMME) project [14]. It is very comprehensive and supports the encoding of philological information. However, it was designed primarily as a file format for a critical edition software application developed for this repertoire and is not widely used outside it.

Many musical codes have been designed together with the development of music notation software applications. Most of these solutions target printed output and the encoding design and rendering tools are tightly bound together. Such is the case for MUP [16] and LilyPond [17]. In LilyPond, the code acts as a set of typesetting instructions to be interpreted by a compiler. This makes it a very powerful and flexible solution for producing a tailored output, including for specific repertoires. It can be used for Renaissance music and is highly customizable [18]. One limitation of LilyPond, however, is that it is very difficult to parse outside its own environment. One reason is that the structure of the data, which can be of arbitrary complexity, is driven more by the desired visual output than by the underlying logical structure of the notation.

Guido is another solution that follows a similar approach where the design of encoding structure is developed hand-in-hand with a rendering software component [19, 20]. It is designed to be embedded into a wide range of environments and is more flexible than LilyPond in that regard. To our knowledge, however, no extension for early music notation is available.

Quite sophisticated rendering engines are embedded in music notation software applications such as Finale [21] or Sibelius [22]. Most of these are so called ‘wysiwyg’ (‘what you see is what you get’) desktop applications. Since many of them are closed-source commercial applications, the musical data representation, both in-memory and in the files, is usually kept undisclosed. Ultimately, what counts for the user is the final (printed) result. For representing uncommon music notation features, users have continuously been tweaking the use of these applications to obtain the desired output. To do so, custom symbols can be used, or the music font can be changed. While this can work for producing printed scores, it is cumbersome to create since the applications are pushed beyond their design scope. Furthermore, it significantly reduces the interoperability of the data. Exporting them, for example to MusicXML, mostly produces musically nonsense data.

Remaining in the commercial domain, software packages such as SCORE [23] can produce highly customizable and fine-tuned output, and have indeed been used for decades by the music publishing industry. The principal drawback in this case is not only the difficulty of using the software, which is a specialized task requiring special training, but also that SCORE does not encode symbolically the music to be typeset. Its input format is focused on the precise description of the elements to print on the score, not what they mean. Extracting simple symbolic data, such as the notes on their own, is a complex and not completely accurate translation process.

4. MEI AND VEROVIO

4.1 Encoding

MEI is a community-driven effort to define a common encoding scheme for describing music notation documents. One the principal goals of MEI is to model music notation and how to represent it digitally in a structured and meaningful way. This approach differs from other initiatives where the goal is to encode the music notation for it to be usable by existing computer software applications, or for it to be typeset. In that sense, MEI acts as an application agnostic music encoding framework for representing music notation documents [24].

One characteristic of MEI is that it is organized into modules. There is an 'MEI' and a 'shared' module for common elements that form the basis of the MEI schema. Then each module groups the definition of the XML elements and attributes of a specific notation or application sub-domain. For example, CWMN and mensural notation are defined in two distinct modules. The advantage of having modules is that the schema definition is kept structured and can be adjusted according to needs. When necessary, valid values for some attributes can be defined differently in two modules, making their intended use more precisely defined and clearer. However, if all modules are activated, all possible values will be valid. The note duration is a good example since it has two distinct data types for CWMN and mensural notation.

Additional modules provide editorial markup capabilities – 'critapp' and 'edittrans'. With editorial markup, it is possible to encode alternative content using a parallel segmentation approach. This means that, when necessary, the encoding tree is divided into two or more alternative sub-trees. This is used for encoding variants between sources in critical editing with <app> and <rdg> elements. In a more generic way, the <choice> element can be used for encoding different representation options.

MEI is application agnostic and aims to be as comprehensive as possible. To this end, different representation domains are defined, although making a clear cut between the different domains in music notation is sometimes impossible. MEI makes the distinction between the visual domain, the logical domain and the gestural domain, the latter referring to how the music notation is expected to be rendered in sound. This separation is quite powerful and makes it possible to encode different domains simultaneously.

4.2 Rendering

Verovio is a music rendering engine written in C++ based on MEI [25]. Its goals are to be small and self-contained, without complicated external dependencies, and to be easily embeddable in other applications. It can work as a standalone tool or as a linked library, or, by compiling it using Emscripten, directly as a JavaScript library for in-browser rendering. This latter option enables it to build rich and responsive web-based music applications. This also relies on the output format, SVG, which greatly facilitates user interaction with the underlying encoding in web-based environments.

The internal data representation of Verovio is based on MEI. This means that music encoded in MEI is not transcoded before being rendered (for example to MusicXML for use in Finale or Sibelius) but is directly interpreted by the rendering engine. The MEI structure is preserved as far as possible in the SVG output of Verovio. This also means that Verovio inherits MEI's modularity and can easily be extended to support the different modules of the specification.

Verovio follows the SMuFL specification for its music font, making it easy to change [26].

4.3 Black notation

Encoding black notation is quite straightforward in MEI. It is encoded with the @colored attribute on the <note> element that indicates coloration, i.e., inverted note heads. The duration encoded in the @dur attribute is expected to be the duration of the corresponding uncolored note. In the Cazzati example of Figure 3, for the first measure, this means a value of "2" for the colored half note and "1" for the colored whole note.

```
<measure n="1">
  <staff n="1">
    <layer n="1">
      <note pname="c" oct="3" dur="2"
        colored="true"/>
      <note pname="f" oct="2" dur="1"
        colored="true"/>
    </layer>
  </staff>
</measure>
```

Figure 8. Black notation in MEI. The <note> element has a @colored attribute that can be set to "true".

The encoding of Figure 3 can be visualised as is with Verovio (Figure 9). The appropriate note heads are selected for the colored notes. For the second note of the first measure, this is the SMuFL code U+E0FA for a filled whole (semibreve) note head (noteheadWholeFilled). This character is not the same as the note head used for quarter notes, as it is specifically designed for coloration.

Figure 9. Black notation with Verovio. The appropriate note head is displayed.

4.4 Void notation

Void notation is more complex because it actually introduces a gap in the scale of note durations in the visual domain. There are indeed no quarter notes in the visual domain of void notation since they are visualised as void eighth notes. However, their actual (sounding) duration is still the duration of a quarter note. This means that void notation introduces a dichotomy between the visual and gestural domains from the quarter notes on. Figure 11 illustrates how the second staff of the second measure of the Charpentier example from Figure 5 can be encoded in MEI. The visual duration of the voided notes is encoded in the @dur attribute, namely with “8”, since they need to be visualised as eighth notes. Their gestural duration is encoded in the @dur.ges with a value of “4”. The voided characteristic is encoded with the @colored attribute.

```
<staff n="2">
  <layer n="1">
    <note pname="f" oct="5" dur="2" dots="1"/>
    <beam>
      <note pname="e" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="d" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="c" oct="5"
        dur="8" dur.ges="4"/>
    </beam>
  </layer>
</staff>
```

Figure 10. Void notation in MEI. Because we have a dichotomy between the visual and the gestural domain, both the @dur and @dur.ges attributes need to be used.

The encoding of Figure 10 can be visualised as is with Verovio, with a correct interpretation of the duration of both the visual (voided eighth note) and gestural (quarter note) domains.

Figure 11. Void notation in Verovio. The appropriate note head is displayed with the duration of the gestural domain.

In some cases, it might be desirable to be able to switch from the original notation to a normalized one. In the case of void notation, one way to do this would be to have the rendering tool being aware of this practice and making it visualise the notes with a @coloration attribute set to “true” by only looking at the @dur.ges attribute value (i.e., by ignoring the @coloration and the @dur attributes). However, this would be a very specific implementation and a more generic solution is highly preferable. One way is to act at the encoding level using a parallel segmentation with a <choice> element. With such an approach, the original void notation is encoded in an <orig> element with the original notation, and a normalised version is encoded in parallel in a <reg> element.

```
<choice>
  <orig>
    <beam>
      <note pname="e" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="d" oct="5" colored="true"
        dur="8" dur.ges="4"/>
      <note pname="c" oct="5"
        dur="8" dur.ges="4"/>
    </beam>
  </orig>
  <reg>
    <note pname="c" oct="5" dur="4" />
    <note pname="d" oct="5" dur="4" />
    <note pname="c" oct="5" dur="4" />
  </reg>
</choice>
```

Figure 12. Alternate encoding in MEI. In some cases, it is desirable to have both the original notation and a normalised version in parallel.

Verovio has a ‘choiceXPathQuery’ option that can be used to select a specific child of the choice element for visualisation. By default, the first child of a choice is selected. In our example, in order to select the normalized version, the option would need to be set to ‘./reg’ for selecting the <reg> element instead of the first child.

Since the exact meaning of void notation is still unclear, and since composers used it quite extensively, it is important to encourage proper modern encoding of it. This not only allows the notational particularities desired by the author to be retained, but will also hopefully facilitate further investigation of the problem using a bigger and more coherent dataset once a proper corpus of modern transcriptions has become available.

5. CONCLUSIONS AND FUTURE WORK

The examples we have shown in this paper focus on uncommon rhythmic notations. They show how unusual features can be represented out of the box with MEI. There are also harmonic specificities in the notation of the time that we would like to cover in the future.

5.1 Figured bass

The most idiomatic notational element in the music from the 17th to the late 18th century is figured bass. Yet some high-end musical typesetters, such as Finale, completely lack support for it: a special font is provided that is inserted as lyrics. This approach not only requires an incredible amount of fine-tuning to the score to obtain an acceptable figured bass, it is also nonsensical as symbolic representation.

Other software, such as LilyPond, have a complete figured bass support, albeit with a complex encoding method. The single figures in LilyPond are not directly attached to any notes, but are a free-form independent voice, which is then superimposed on the music notation. While this solution works very well for typesetting music, in spite of the complexity of inserting the single figures, it is very difficult actually to associate the numbers with the notes to which they are attached, making the system useless as a symbolic representation of music plus figured bass.

To complete the support for 17th century music a module for figured bass in MEI is to be proposed. The purpose will be to have a complete representation so the single figures can be analysed in relation to the notes to which they are associated, if needed, and to provide high-quality typesetting output.

5.2 Scordatura

Virtuoso violin music from the 17th century often employs a technique called *scordatura*, in which the single strings of the instrument are tuned to other notes.

Different methods were used to write down music requiring *scordatura*, and most often the written notation marks the notes the player would have performed on a normally tuned string. In this way the hand position does not change in respect to the written note, however the note sounds different from what is actually written. It is a technique somehow in-between tablature and normal notation.

The challenge to typeset *scordatura* is that the written notes no longer correspond to the sounding notes. With commonly available notation systems there is no way to obtain at the same time sounding pitch and written pitch. It is necessary to encode each separately.

5.3 Conclusions

Many specialist software notation packages exist, and a number of these have been developed with specific requirements in mind. Unfortunately such specialized software is often unavailable outside large editors or projects, putting them out of the reach of many researchers. Moreover specialized software often requires special training to use, which can be an unachievable hurdle to master. Lastly a plethora of different systems encodes its data in a plethora of different manners, often not interchangeable between one and another. This impedes theconstitution of large and accessible collections of encoded music, as no shared standard poses an obvious barrier to anyone not using the particular software created for a particular collection.

As illustrated in this paper, the separation of different representation domains offered by MEI together with a tool that can take them into account is the perfect basis for encoding at the same time the visual information (the note position) and the gestural one (the sounding pitch).

Encodings such as MusicXML strive to achieve compatibility across systems, but were created on purpose with no support for ancient music. On the other hand MEI proposes to be a unique container for all western notation styled – extensible to what is not currently supported – with the same compatibility across systems attempted by MusicXML.

Acknowledgments

This work was supported by the Swiss National Science Foundation (SNSF) and the Swiss Academy of Humanities and Social Sciences (SAHS).

6. REFERENCES

[1] J. Grier, The Critical Editing of Music, Cambridge University Press, Cambridge, 1996.

[2] Giovanni Maria Bononcini, *Sonata Ottava* from *Sonate da chiesa à due violini*, Op. 6 (Venezia: Magni, 1672).

[3] A. C. Yeung, "A study of late seventeenth-century compositional practice: The "Sonate da Chiesa" (Opus 6) of G. M. Bononcini", PhD diss., Columbia University, 1994.

[4] Maurizio Cazzati, *Messa Concertata* from *Messa e Salmi a Quattro voci,* Op. 14, (Venezia: Vincenti, 1653).

[5] Masses by Giovanni Pietro Finatti, Maurizio Cazzati, Giulio Cesare Arresti; edited with an introduction by Anne Schnoebelen (Seventeenth-century sacred music; v. 6), p. 103.

[6] S. Thompson, "Once more into the void: Marc-Antoine Charpentier's croches blanches reconsidered", Early Music XXX (1), pp. 83-93, 2002

[7] C. Churnside, "Music printing in early eighteenth-century Bologna: the case of Giuseppe Antonio Silvani and Pirro Albergati (1716-1717)", in *Fonti Musicali Italiane*, no. 17, pp. 105–34, 2012.

[8] G. Giovani, “Un capitale vivo e morto ad uso di samparia. Nuovi documenti sull'editoria musicale bolognese del Seicento”, in Giorgio Monari (ed) *Studi sulla musica dell’età barocca*, Libreria Musicale Italiana, Lucca, 2012

[9] R. Zitellini, “A study on the auto-revision activities of Maurizio Cazzati in his music reprints”, Schweizer Jahrbuch für Musikwissenschaft, serie 33 (2013), Peter Lang, Bern 2016.

[10] Pirro Capicelli Albergati (Bologna: Giuseppe Antonio Silvani, 1717).

[11] H. van der Linden, “Early eighteenth-century music type for the printer Giuseppe Antonio Silvani: Correspondence and other documents”, Tipofilologia, nr. 5, pp. 27–68, 2012

[12] T. G. Merrill, Canticle in Honor of Saint Xavier. <http://www.sscm-wlscm.org/main-catalogue/browse-by-composer/45-wlscm-no-13>

[13] E. Selfridge-Field (Ed.), *Beyond MIDI: The Handbook of Musical Codes*. The MIT Press, Cambridge, 1997.

[14] MakeMusic, *MusicXML*. <http://www.musicxml.com>

[15] T. Dumitrescu and M. van Bechum, *Computerized Mensural Music Editing*. <http://www.cmme.org>

[16] Arkkra Enterprises, *Mup*. <http://www.arkkra.com>

[17] H. W. Nienhuys and J. Nieuwenhuizen: “LilyPond, a system for automated music engraving,” *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pp. 167–72, 2003.

[18] LilyPond, *Typesetting mensural music*. <http://lilypond.org/doc/v2.18/Documentation/notation/typesetting-mensural-music.en.html >

[19] H. Hoos and K. Hamel, *The GUIDO Music Notation Format*. <http://guidolib.sourceforge.net/doc/GUIDO-Music%20Notation%20Format.html>

[20] C. Daudin, D. Fober, S. Letz and Y. Orlarey, “The Guido Engine: A toolbox for music scores rendering,” in *Proceedings of Linux Audio Conference*, 2009, pp. 105–11.

[21] MakeMusic, *Finale 2014* <http://www.finalemusic.com/>

[22] Avid, *Sibelius*. <http://www.avid.com/en/sibelius>

[23] San Adreas Press, *SCORE*. <http://www.winscore.info/>

[24] A. Hankinson, P. Roland and I. Fujinaga. “The Music Encoding Initiative as a Document-Encoding Framework,” in *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pp. 293–8.

[25] L. Pugin, R. Zitellini and P. Roland, “Verovio: A library for engraving MEI music notation into SVG,” in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014),* pp. 107–12.

[26] Steinberg, *SMuFL*. <http://www.smufl.org>