Gioqoso, an online Quality Assessment Tool for Music Notation

Francesco Foscarin, David Fiala, Florent Jacquemard, Philippe Rigaux, Virginie Thion

GIOQOSO, AN ON-LINE QUALITY ASSESSMENT TOOL FOR MUSIC NOTATION

Francesco Foscarin
CNAM, Paris
first.last@lecnam.net

David Fiala
CESR, Tours
first.last@unvi-tours.fr

Florent Jacquemard
INRIA, Paris
first.last@inria.fr

Philippe Rigaux
CNAM, Paris
first.last@lecnam.net

Virginie Thion
IRISA, Rennes
first.last@irisa.fr

ABSTRACT

Quality is a daily concern to everyone involved in the production of digitized scores. We propose an on-line interface devoted to music notation, freely accessible to the community, intended to help users to assess the quality of a score thanks to a combination of automatic and interactive tools. This interface analyzes a score supplied in MusicXML or MEI, and reports quality problems evaluated with respect to a taxonomy of quality rules. We expose the motivation, describe the interface, and present the methodology.

1. INTRODUCTION

It is a common and shared experience that achieving high quality standards for digitized scores is quite difficult and currently requires a lot of time and efforts devoted to inspect the score rendering and detect mistakes. The difficulty of this task is due to the complex semiology of music notation. The issue is particularly sensible in the context of collaborative editing, since each contributor is free to use her own engraving software, and to communicate with others via some XML format, typically MusicXML [1], sometimes MEI [2, 3], and probably in a near future the W3C MNX format [4].

Unfortunately, these XML-based encodings are extremely permissive, and allow for all kinds of problems regarding correctness, consistency and completeness. This can be understood if we consider that they have to adapt to the wide flexibility and variability of music notation throughout ages. This is also probably unavoidable, given the complexity of rules that can hardly be expressed as constraints in the document's schema. As a result, music score encoding a quite error-prone, and currently requires a careful revision by human experts as part of the publishing process.

1.1 Evaluating a score

Quality evaluation is commonly done by a combination of audio and visual inspections. Given the high semiologic complexity of music notation, this evaluation cannot be fully reliable, and would highly benefit from the assistance of automatic tools. What makes things even worse is that even if a score has been checked visually by several people who spent hours to inspect every detail, this does not guarantee that the underlying encoding is correct. Let us take two simple and concrete examples:

1. *Lyrics encoding*. The association of text and music obeys some complex rules. Lyrics are decomposed in syllables, and, at the graphical level, syllables from a same word are linked by dashes, melismas are indicated by underscores, *etc.* People engraving music have to be aware that a correct encoding has to distinguish the syllables from the metadata that describes how they are interrelated and linked to the music. We already found many examples where both aspects are glued, because the engraver directly encodes continuation symbols in the text itself. As a consequence, although not directly visible, the score encoding becomes faulty: the text cannot be cleanly extracted or searched, and some notes in melismas are not properly attached to syllables.
2. *Layers encoding*: many music pieces are organized as a combination of layers, and in order to make sense of these pieces, it is important that the engraver identifies the layers content and carefully reflects them in the encoding. Unfortunately, many engravers loosely use the layer concepts in engraving softwares for tricking the visual rendering, losing the internal music structures.

We can cite many other examples where an apparently correct score, at least when printed or rendered on a screen, turns out to be wrongly encoded internally: slurs instead of ties, title or composer entered as raw text, and not as metadata, *etc.* This results in unexpected distortions when another renderer is used, and makes the music representation unsuitable for other usages: analysis, audio/score alignments, or production of alternative representations (Braille for instance).

1.2 Defining and measuring quality

Defining and measuring the quality of a score encoding is not easy. First, there is no universal definition of what a "correct" score encoding is: it highly depends on the music itself on one hand, and on the score usage on the other hand.

Second, many abstraction levels can be considered, and many granularities. Some aspects are purely syntactic (do all slurs have a start/end point? Are all measures exactly filled?), other pertain to metadata, which may or may not be mandatory (title, composer, date, copyright). Some aspects are specific to the score layout (symbol overlapping, appropriate position of clefs and staves). And, of course, the music content itself has to be correct and should faithfully reflect the source and editors choices. The latter is probably the most difficult part to assess with an automatic evaluation, although we can imagine to check if the material is consistent with the style and expected idiomatic features.

All these points have to be simultaneously taken into account by a proof-reader. As explained above, visual inspection is both unreliable and insufficient, in particular if we are keen to ensure an accurate representation of the score content, apt at being exploited in other contexts that the mere printing of the music sheet. Controlling manually the encoding itself is not really an option, even assisted by advanced editors – A single inspection of a large XML file should be enough to be convinced that nothing can reliably done at this level. What we need is a holistic approach that combines visual and audio evaluation with an automatic inspection of the encoding to report potential quality issues.

1.3 Our approach

We propose a tool that attempts to provide in a single interface all the components that participate to a score evaluation, and makes this evaluation automatic as much as possible. This tool is publicly available online [1] and can be used by anyone to evaluate an XML-encoded score (MusicXML or MEI) as soon as the document can be retrieved from some URL. The implementation is, and will continue to be, in progress, because the list of quality rules that can be envisaged is potentially endless. However, we believe that the foundations of our method are now established, and that the main functionalities of the user interface are operational. We therefore submit to the TENOR community the current status of our work. The main contributions are:

1. A taxonomy of quality rules that relies in particular on a distinction between the concepts of *score content* and *score engraving*. This distinction was proposed in one of our earlier works [5] as a necessary step to make sense of the heterogeneous information gathered in digitized scores. It is used as the foundation of a hierarchical presentation of quality aspects which, in our opinion, helps the end user to organize her evaluation.

2. An implementation of representative quality rules for each of the main categories of our taxonomy. We describe a sample of indicators to illustrate their specific features.

3. Last but not least, an integration of the methodology in the GIOQOSO public Web interface.

For the sake of concreteness, we start with a description of the user interface in Section 2. Section 3 explains the foundations of our digitized scores quality model. We examine our taxonomy and some representative examples in Section 4 and conclude the paper in Section 5.

2. THE GIOQOSO ONLINE INTERFACE

Figure 1 shows the current status of the GIOQOSO tool[1]. GIOQOSO is integrated in the NEUMA Digital Score Library [6], but is an independent component that can be used to analyze any XML score accessible at a public URL.

2.1 Importing and displaying the score

Figure 1 illustrates how we import a score coming from the *Lost Voices* CESR project [2]. The score has to be encoded either in MusicXML or in MEI. However, when the input is in MusicXML, an internal conversion is operated first to obtain an MEI encoding that enjoys two major advantages in our context.

1. Each element of the score (notes, rests, slurs, measures, staves, *etc.*) has a unique id. This is is essential to *annotate* this element with some semantic label, in our case, a quality indicator. For instance a note can be annotated with a *missing lyrics* indicator, or a measure with a *incomplete duration* indicator.

   MusicXML, unfortunately, does not offer this ability to refer to score elements. This is one of the main new features that will be incorporated in the forthcoming MNX standard [4].

2. A second advantage of the MEI encoding is that it comes with several analysis and interactive tools. We use in particular the Verovio toolkit [3] [7] to display and interact with the score. Verovio relies on a conversion from MEI to SVG that preserves the id of elements. As a result, an annotation (*i.e.*, some meaning attached to a note or a measure) can be graphically displayed as a decoration of the corresponding SVG element.

The ability to play a MIDI rendering of a score, possibly starting from any note, is also a Verovio feature. This functionality corresponds to the standard "Play" option proposed by all score engravers, and is the quite useful tools when it comes to check the content of a score.

2.2 Showing/hiding quality annotations

The document is analyzed on-the-fly in order to complete it with quality annotations. Each annotation is an instance

[1] http://neuma.huma-num.fr/quality

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

[3] http://verovio.org

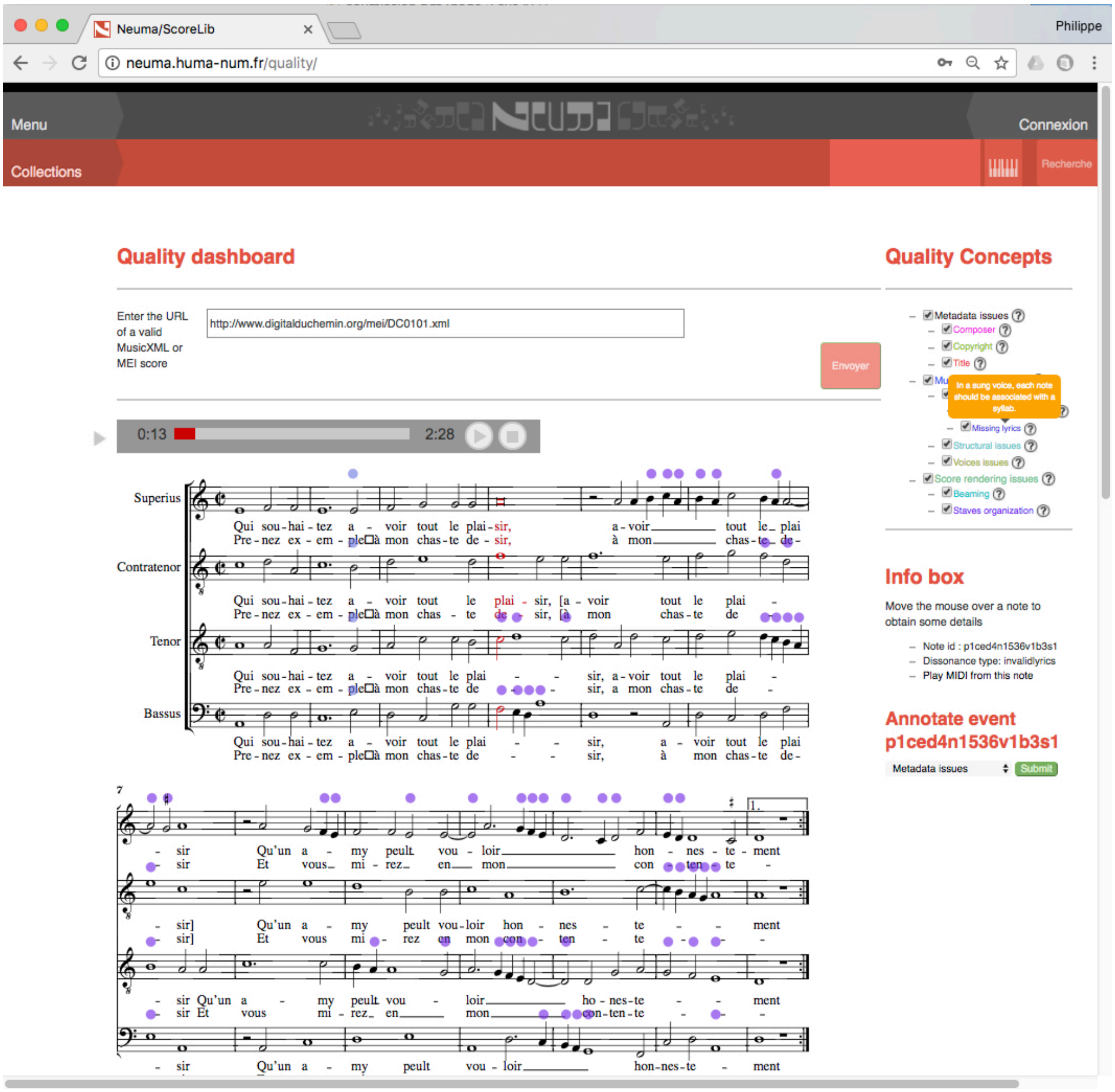

Figure 1. The GIOQOSO User Interface

of a quality indicator, and the indicators themselves are organized as a forest, displayed in the top-right part of the user interface (see also Section 4).

The taxonomy of the quality model is extensible. We add new rules regularly, based on input from our scientific experts (the CESR and IReMus musicology labs), on best notational practice found in reference sources on score rendering/engraving, *e.g.* [8], and on mere exploration of various online score libraries that reveal many encoding and rendering issues.

In the interface, each indicator comes with a description that can be highlighted by dragging the mouse over its name (the orange rectangle in Figure 1, column 'quality concepts'). Every annotation is displayed as a small colored circle above the elements or groups of elements that constitute the annotated fragment. Its color characterizes a specific quality indicator. The user can hide/show a set of annotations by clicking on any level of the model tree. This makes convenient to focus on a particular aspect, or to ignore altogether some indicators if they are deemed irrelevant.

2.3 Interactions

Finally, actions can be undertaken by the user. Each annotation can be inspected in detail by clicking on it. The *Info box* part of the interface then displays details on the related score elements, and on their annotations (there might be many). A form is also proposed to report an annotation error, or to complete existing annotations. Such inputs might become quite useful in the future in order to include user feedback in the context of a large collaborative system.

Note that since the score is loaded from its remote location, the user can directly correct the identified issue on her local version. It suffices then to reload GIOQOSO to trigger a new evaluation of the quality rules that will hopefully show that some formerly identified quality issues have been fixed. GIOQOSO can therefore be seen as a complementary tool closely and easily integrated to the user's score production environment. The only requirement is that the score under production is accessible at a fixed URL.

3. MODELING DIGITIZED SCORES QUALITY

Our model of rules for notational quality follows a conceptual view of score of score production that distinguishes three steps (Fig. 2).

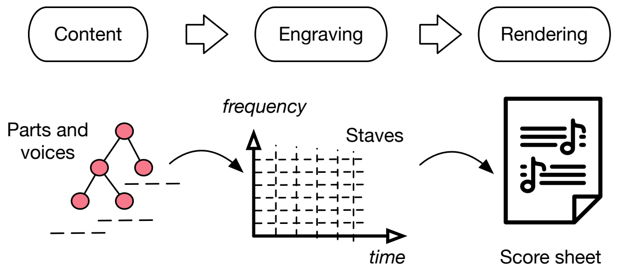

Figure 2. The workflow of (digitized) score production

1. *Score content modelling*. This part covers all aspects related to what we call the *score content*, independently from any encoding or rendering concern. Essentially, it captures the *structural* organization of a score in parts and *streams* [9], and the description of streams as time-dependent elements.

2. *Score engraving*. Score engraving denotes the mapping of the score content into a set of staves. We model a staff as a grid covering a restricted range in the space of frequencies, and the mapping associates a content with a 2D (frequency, time) space.

3. *Score rendering*. The final steps take a score content, score engraving specifications, and produces a layout of score based on the properties of a specific media (paper, screen, *etc*).

We believe that this distinction is extremely useful to identify and characterize the specific quality issues that can occur at each step, and to determine how we can evaluate and possibly fix these issues.

First, clearly, the last step (score rendering) depends on the rendering software and on the properties of the displaying media. Therefore, we consider this part as out of scope for the score quality evaluation process: a high quality score can be displayed very badly with a poor renderer or on a tiny screen.

This leaves us with the distinction between *score content* and *score engraving*. We think that it makes sense for exactly the same reasons that led to separate the content of web pages (structured in HTML) from their display features (defined with CSS rules) [4]. Defining the *content* of a score, and evaluating its quality, is a data modelling and representation task. It requires the definition of the structure of a score, and the specification of constraints on instances of this structure. On the other hand, *engraving* is a process that applies to a score content, and defines the relationships between this content and a 2-dimensional space organized with respect to a temporal dimension (abcissa) and a frequency dimension (ordinate). Evaluating the engraving quality implies to take into account both the content and the mapping.

3.1 The score content model

The "score content" focuses on the aspects of a digital score representation that describe the intended production of sounds, and is independent from any visualization concern. If we assume an ideal music performer, the content is the part of the score that contains the sufficient and necessary information to produce the intended music. In order to decide whether a piece of data belongs or not to the content, we just have to wonder whether it is likely to influence this music production. A MIDI player is a possible candidate, but we actually require a more sophisticated performer model, apt at taking account for instance of the meter to infer strong and weak beats.

In an earlier work, we proposed a *notation ontology*, called MusicNote [5], to model this content [10]. Essentially, a score is modeled as a *hierarchical structure*, where leaves consist of *streams*, and inner nodes of *parts*. A stream is a sequence of *events*, which can belong to several subclasses. Let us explain the structural aspect first by taking as an illustration the sketch of a piano concerto score (Fig. 3).

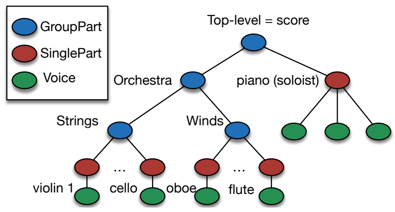

Figure 3. Structure of a score

The score is made of *parts*, where the concept of part is is refined into two sub-concepts. A *group* (of parts) consists of a set of subparts, and mostly serves the organizational aspect of the score. For instance, the orchestral material of a concerto score typically defines a group for wind instruments, another one for string instruments, *etc.* A *single part* encapsulates the music events assigned to an individual performer (instrument or vocal). Fig. 3 shows for instance a single part for the soloist (piano), another one for the violins, cellos, *etc.* The informations related to measures (in particular time signatures) are represented at this level. A single part contains one or several *streams*.

Streams are objets where music content, as time-dependent production of sounds, is actually described, as illustrated by Fig. 4. A *stream* is essentially a time series of *events*, where an event denotes the production of a sound artifact at a specific timestamp (the "onset"). Particular cases of events are notes and chords (with pitch and duration information), textual contents, or information on dynamics and articulation.

[4] The metaphor also holds for the *rendering step*, carried out in the case of HTML by a Web browser that adjust the textual content and CSS rules to the displaying window.

[5] http://cedric.cnam.fr/isid/ontologies/files/MusicNote.html

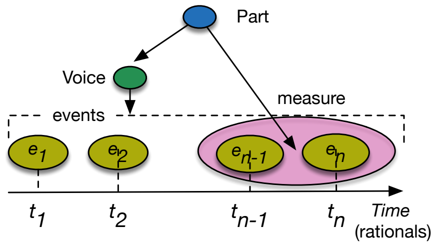

Figure 4. Stream as a time series of events

The quality issues that related to the score content concept are therefore organized with respect to the above ontology [6] .

3.2 The score engraving model

A score is a graphical artifact that represents some music content according to two dimensions:

1. *Time*. This dimension is represented by the horizontal axis, and is discretized in measures, beats, and finite subdivisions of beats.
2. *Frequencies*. Sound frequencies are represented on a vertical axis, and discretized in octaves, and subdivision of octaves in (usually) 12 semi-tones.

This yields a 2-dimensional discretized space, that could be represented as a grid. In principle, a score could be fully displayed in this grid, each note being a segment whose height corresponds to its frequency, and length to the note duration. The score engraving is close to this general model, but makes some choices, motivated by practical reasons, that lead to the usual layout. First, each part (or instrument) gets its own space visualization in order to avoid the confusion that would result from the merge of several parts with similar ranges in the same layout. Second, since the range of a single instrument is usually restricted, the frequency grid allocated to this instrument is reduced to a few lines that cover this range, of *staff*. The common representation chooses to use 5 lines, and to encode the range with a clef that gives the frequency of one of those lines (*e.g.*, the second line for treble clef).

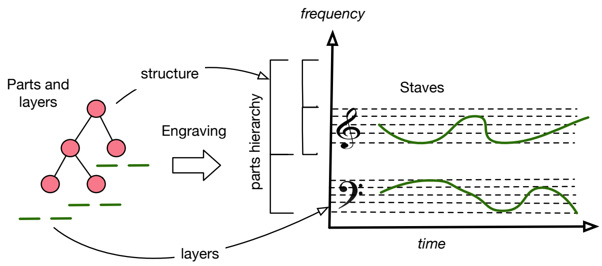

Figure 5. Engraving = mapping the content to (time, frequency) space

[6] In some cases, these issues can even be formalized as rules expressed over the ontology with SWRL, the Semantic Web Rule Language [11]. We refer to [10] for a discussion on the pros and cons of a declarative approach to specify annotations semantics.

This perspective on score engraving is summarized by Fig. 5. The engraving rules take a score content, determine the number of staves, allocate parts to staves, and develop the stream representation on each staff.

Our quality model relies on this perspective, and focuses on the organization of staves, their relationships, and on the inner quality of stream representation for each staff. The general question that we try to address in this context is: to which extent the content/staves mapping defined by the engraving ensures a consistent and correct layout of score? If the engraving quality is high, then we can expect that a good renderer will be able to produce a readable score display at visualization time.

3.3 Metadata

Finally, we consider a third, optional part of score encoding: metadata. Metadata is data about data, *i.e.*, in our case, any content that annotates either the score content or the score engraving. The title, subtitle, composer are metadata that annotates a score as whole. Instrument names annotates parts. There are at least two reasons to incorporate metadata issues in quality evaluation. First, metadata supplies in some cases some knowledge which is useful to measure a quality indicator. Knowing the instrument for a part allows for instance to check that the range of the music content is compatible with this instrument, or that the clef is appropriate. Second, metadata is typically a factor of inconsistencies when we consider quality concerns at a collection level. Music collection editors are eager to ensure that the level, accuracy and encoding of metadata are similar for all the scores. Although the present paper focuses on single scores, this motivates the inclusion of metadata as part of our quality model.

4. THE TAXONOMY

Based on the models introduced in the previous section, we created a taxonomy of quality indicators. The taxonomy is a forest where each tree corresponds to a "facet" of quality evaluation, and contains the related set of indicators. Currently, our taxonomy contains three such trees. It is fully accessible at http://neuma.huma-num.fr/quality/model, and partially described below. Quality indicators in boldface are detailed in the following as representative examples of the salient categories.

1. Score content issues
   - (a) Structural issues
     - i. Unbalanced parts
   - (b) Stream issues
     - i. Pitch
       - A. Out of range
     - ii. Rhythm
       - A. **Incomplete measures**
       - B. Overflowing measures
     - iii. Lyrics
       - A. **Missing lyrics**

B. Invalid lyrics encoding

2. Engraving issues

   (a) Staves organization

      i. Invalid staff order

      ii. Too many parts per staff

   (b) Staff parameters

      i. Invalid key signature

      ii. Invalid clef

   (c) Staff layout

      i. **Erroneous Duration**

      ii. **Unappropriate beaming**

3. Metadata issues

   (a) Missing title

   (b) Missing composer

   (c) Invalid instrument name.

   (d) ...

Note that we chose to organize our taxonomy with respect to *functional* concepts that are highly specific of the data at hand. Another possible organization would consist in considering generic quality problems [12] such as completeness, accuracy, and consistency. We believe that, in essence, quality is a multi-dimensional problem. The choice to favor the functional dimension is motivated by the need to help the user focusing on a "semantic" perspective during her inspection of the score. As such, the hierarchical organization mostly serves to navigate in the rules trees to hide/show some of them.

4.1 Score content issues

4.1.1 Structural issues

As an example of structural quality indicator, we check that all single parts have the same length. This is done by computing the sum of the durations of all the events in streams and comparing.

For this purpose, we rely on a routine of MUSIC21 [13] for extracting the duration of every event, expressed in fraction of the duration of a quarter note (quarter length). The correspondence between this duration value and the notated duration value (in term of note figures) is checked separately in GIOQOSO, see Section 4.2.2.

4.1.2 Music notation issues

At the stream level, an important property is that all the measures are correctly filled, *i.e.* that for each measure, the total duration of the events contained corresponds to the expected measure length, according to the time signature (specified in the embedding part). This is done using the same Music 21 duration event information as above.

Some issues related to lyrics quality have already be mentioned in introduction.

4.2 Score engraving issues

4.2.1 Staff parameter issues

This part of the taxonomy covers quality problems related to an incorrect or inconsistent assignment of parts to the staves system and on the parameters that dictates how the music content is rendered on a staff. The following is a list of examples that related this "functional" approach to some common quality dimensions [12].

1. *Consistency*. We check that all key signatures are consistent, including a correct transposition for transposing instruments. This is simply done by checking the key signatures encoding of all the parts in the XML document.

2. *Correctness*. The clef should be chosen to ensure that the majority of notes lies inside the staff's range (i.e., do not show a bass part on a treble claf staff).

3. *Completeness*. We check that all parts of the score are assigned to a staff, with a maximum of two parts per staff.

4.2.2 Staff layout issues

In music theory, there are precise rules for deducing actual durations from note values and meter (TS) and common practice / recommendations for writing rhythms (using in particular beams for defining nested groups), in order to improve score readability and emphasize the meter.

Digital scores *e.g.* in MusicXML usually contain rhythmic elements of different nature: features related to score content, like time signature and actual note durations, and features related to engraving content, like note symbols and beams. Despite their strong relationship, these elements can be presented independently in documents. This redundancy can be source of inconsistency in rhythm notation.

Let us give below some details about the procedures proposed in GIOQOSO for assessing the quality indicators related to the rhythmic notation in scores. That concerns in particular the consistency of the different elements represented durations and the satisfaction of some beaming conventions.

Our approach works by extracting tree structures from a score XML document and then performing verification on these trees. We consider a hierarchical model of rhythm notation inspired by the Rhythm Trees of Patchwork and OpenMusic [14, 15, 16, pages 976-978]. However, our model differs from RT in several aspects: In addition to the representation of proportional durations of notes, it also includes engraving elements related to rhythm notation (note figures, beams, *etc*). Let us describe more precisely this representation on the example in Figure 6.

Every leaf represents a note (or rest, chords...), with a label describing the note figure: `n` for simple note head, `n..` for double-dotted note head, `-n` for a note tied to the previous one... Every node is associated a duration in quarter length: The root node is associated the duration of a whole measure and every edge in the tree is labeled by the ratio between the duration of the parent node and the child

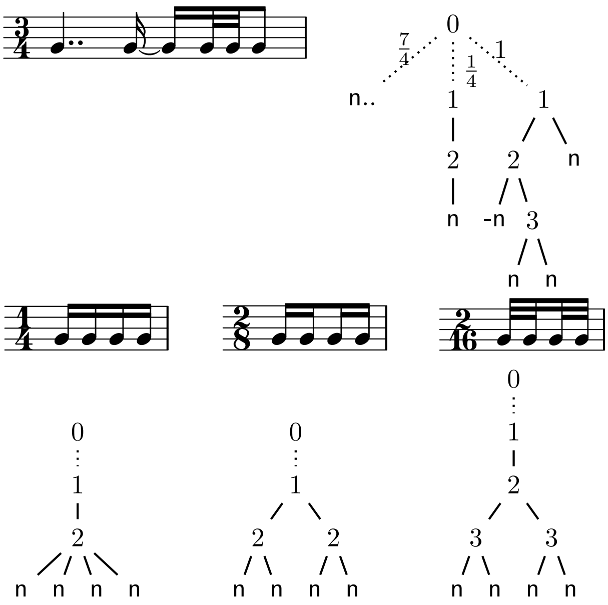

Figure 6. Beaming-tree representations of 4 measures.

node. For readability, we omit the edge labels when all sibling have same ratio. Finally, the representation of beams follows these two principles:

1. every leaf represents a note (or rest) whose number of beams is the depth minus 1 (in Figure 5, inner nodes are labeled by their depth).

2. the number of beams between two successive notes is the depth of their least common ancestor node (which is not necessarily their direct parent).

The first property holds both for isolated notes, like the two first notes in the 3/4 measure in Figure 5, and notes in groups. And, according to the property 2, two notes not connected by beams are child of the root note (see the same example 3/4 as before).

In GIOQOSO, we extract a tree as above for each measure in a digital score. The structure of trees is inferred from the beaming information and note types (as specified *e.g.* by the MusicXML element `type`) in the digital score file (MusicXML element `beam`), and the edge labels are computed from the durations given in the file (MusicXML element `duration`).

Then, several properties are checked on the trees. For instance, we check the consistency between note durations (quarter length) and the note types (a note type depends on a leaf label, a number of beams computed as above and the arity of inner nodes representing tuplets). A detected inconsistency can be seen as a critical issue in a score file, that may result in many errors when processing the score.

We also check some beaming conditions, less critical but that help the readability of the score. For instance, some position corresponding to strong beats in measures (like the third beat in a 4/4 measure) should not be crossed by beams, see [8] page 155. This can be checked using the property (2) above (in that case the depth of the least common ancestor of last note before position and first note after must be 0).

For other readability motivations, big groups of short notes are easier to read when subdivised in subgroups whose duration depends on the meter, providing that: *the number of beams separating the groups is equal to the duration of the groups they separate*, see [8]. This is illustrated in Figure 7 and can also be checked using the property (2) above. Failure when checking such properties can be signaled as recommendations in GIOQOSO.

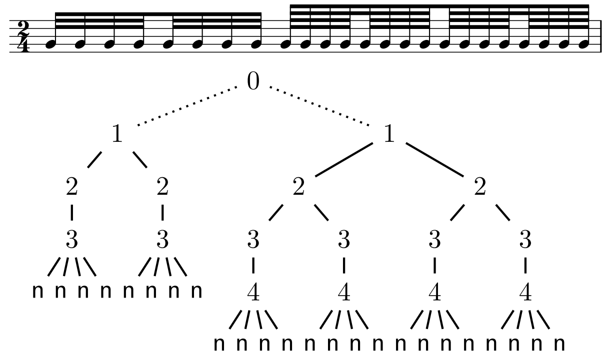

Figure 7. Groups of inner beams of various durations [8].

5. CONCLUSION

In this paper, we proposed a methodology for assessing data quality of digitized scores. We believe that the topic is important, because scores can no longer be considered as mere graphic artifacts, but as digital pieces of information. Assessing the quality of this information is essential, not only for a proper rendering on various media, but also for preserving, exchanging, and analyzing score content in all kinds of future applications.

We hope that our approach provides a ground for proof-checking score beyond graphical concerns. It requires of course several extensions in the future to fully achieve its goals. First, the list of quality indicators currently evaluated is by no means complete, and we can bet that it will never be. This is essentially harmless, this the design of our methodology makes it easily extendible. Second, we currently focus on single score inspection. In the context of collections and digital score libraries, the consistency of the encoding choices for all the scores of the collection is essential. This is particularly sensible for metadata that should be uniformly handled.

Finally, a part a score proof-reading which is basically left apart for the moment is the correctness of the content itself with respect to the source. There is no easy solution to the problem, and it appears that we will remain dependent on the user's expertise for this matter.

Acknowledgments

This work is partially supported by the GIOQOSO project [7], and the ANR MuNIR project [8].

[7] http://gioqoso.irisa.fr/

[8] http://cedric.cnam.fr/index.php/labo/projet/view?id=41

6. REFERENCES

[1] M. Good, *MusicXML for Notation and Analysis*. W. B. Hewlett and E. Selfridge-Field, MIT Press, 2001, pp. 113–124.

[2] P. Rolland, "The Music Encoding Initiative (MEI)," in *Proc. Intl. Conf. on Musical Applications Using XML*, 2002, pp. 55–59.

[3] "Music Encoding Initiative," http://music-encoding.org, Music Encoding Initiative, 2015, accessed March 2017.

[4] "W3C MNX Proposal Overview," https://w3c.github.io/mnx/overview/, 2017, last accessed May 2017.

[5] S. Cherfi, F. Hamdi, P. Rigaux, V. Thion, and N. Travers, "Formalizing Quality Rules on Music Notation âĂŞ an Ontology-Based Approach," in *Intl. Conf. on Technologies for Music Notation and Representation (TENOR'17)*, 2017.

[6] P. Rigaux, L. Abrouk, H. Audéon, N. Cullot, C. Davy-Rigaux, Z. Faget, E. Gavignet, D. Gross-Amblard, A. Tacaille, and V. Thion-Goasdoué, "The design and implementation of neuma, a collaborative digital scores library - requirements, architecture, and models," *Int. J. on Digital Libraries*, vol. 12, no. 2-3, pp. 73–88, 2012.

[7] L. Pugin, R. Zitellini, and P. Roland, "Verovio: A library for engraving mei music notation into svg," in *ISMIR*, 2014, pp. 107–112. [Online]. Available: http://verovio.org

[8] E. Gould, *Behind Bars*. Faber Music, 2011.

[9] E. Cambouropoulos, "Voice and stream: Perceptual and computational modeling of voice separation," *Music Perception: An Interdisciplinary Journal*, vol. 26, no. 1, pp. 75–94, 2008.

[10] S. Cherfi, C. Guillotel, F. Hamdi, P. Rigaux, and N. Travers, "Ontology-Based Annotation of Music Scores," in *Intl. Conf. on Knowledge Capture (K-CAP'17)*, 2017, austin, Texas, Dec. 4-6 2017.

[11] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosofand, and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," W3C Member Submission, W3C, 2004. [Online]. Available: http://www.w3.org/Submission/SWRL/

[12] L. Berti-Equille, I. Comyn-Wattiau, M. Cosquer, Z. Kedad, S. Nugier, V. Peralta, S. S. Cherfi, and V. Thion-Goasdoué, "Assessment and analysis of information quality: a multidimensional model and case studies," *IJIQ*, vol. 2, no. 4, pp. 300–323, 2011. [Online]. Available: http://dx.doi.org/10.1504/IJIQ.2011.043780

[13] M. S. Cuthbert and C. Ariza, "music21: A toolkit for computer-aided musicology and symbolic music data," in *ISMIR*. International Society for Music Information Retrieval, 2010.

[14] M. Laurson, "Patchwork: A visual programming language and some musical applications," Sibelius Academy, Helsinki, Tech. Rep., 1996.

[15] C. Agon, K. Haddad, and G. Assayag, "Representation and rendering of rhythm structures," in *Proceedings Second International Conference on WEB Delivering of Music (CW'02)*, Busch, Arnold, Nesi, and Schmucker, Eds. Los Alamitos: IEEE Computer Society, 2002, pp. 109–113. [Online]. Available: http://dl.acm.org/citation.cfm?id=794192.885321

[16] G. Mazzola, *The topos of music: geometric logic of concepts, theory, and performance*. Birkhäuser, 2012.