



**HAL**  
open science

## Formalizing quality rules on music notation. An ontology-based approach

Samira Si-Said Cherfi, Fayçal Hamdi, Philippe Rigaux, Virginie Thion,  
Nicolas Travers

### ► To cite this version:

Samira Si-Said Cherfi, Fayçal Hamdi, Philippe Rigaux, Virginie Thion, Nicolas Travers. Formalizing quality rules on music notation. An ontology-based approach. International Conference on Technologies for Music Notation and Representation - TENOR'17, May 2017, Coruna, Spain. hal-02475753

**HAL Id: hal-02475753**

**<https://hal.science/hal-02475753>**

Submitted on 21 Jun 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# FORMALIZING QUALITY RULES ON MUSIC NOTATION – AN ONTOLOGY-BASED APPROACH

Samira Cherfi, Fayçal Hamdi, Philippe Rigaux

CNAM

firstname.lastname@cnam.fr

Virginie Thion

IRISA

virginie.thion@irisa.fr

Nicolas Travers

CNAM

nicolas.travers@cnam.fr

## ABSTRACT

We address the issue of expressing and evaluating quality rules on music notation. Since music engraving is a highly flexible process that can hardly be constrained by universal principles and rules, score production still heavily relies on the user expertise in order to make context-dependent decisions. We therefore propose a quality management approach based on a formal modeling of this expertise. We show how to use such a model to express context-aware rules that can be evaluated either *a priori* to prevent the production of faulty notations, or *a posteriori* to assess quality indicators regarding a score or a corpus of scores. The paper proposes a simple ontology for musical notation, shows how quality rules can be formally stated and evaluated, and illustrates the approach with examples drawn from a large digital library of scores.

## 1. INTRODUCTION

Music production is now strongly assisted by sophisticated and powerful computer softwares. They allow to combine all the elements of the notation language, and can in most cases make appropriate decisions regarding quality-sensitive aspects such as, e.g., layout or spacing. We could therefore expect that computer technology would guarantee the production of high-quality scores, validated with respect to a set of well-accepted engraving principles that constrain the notational language [1].

### 1.1 Quality issues

However, it is well-known that this language is highly flexible, due to many cultural and historical contexts where each one presents their own idiosyncrasies. There is an inherent, context-dependent freedom in the adjustment of the common graphic and symbolic elements that constitute a specific score, and this prevents the enforcement of even the most widely accepted principles which can turn out to be inappropriate in some specific cases. To take one example, transcribing a manuscript, in particular for ancient music, raises trade-off issues between the necessity to preserve the original intent of the author, and the adaptation of

handwritten notation to the custom knowledge of today's performers.

The authors of the present paper have been confronted with the need to address issues related to the consistent production of high-level quality corpora encoded in XML-based formats (i.e., MusicXML [2] or MEI [3, 4]), and had to deal with the poor support offered by existing tools. The current, ad-hoc solution adopted so far is to publish a booklet of *editorial rules* prior to the production of the corpus scores. They often take the form of a textual, informal document that enumerates guidelines regarding the encoding of music and helps the editor(s) to find a consistent approach balancing the need to both preserve the sources and to deliver a consistent material to users (musicologists, performers, librarians) who access the corpus.

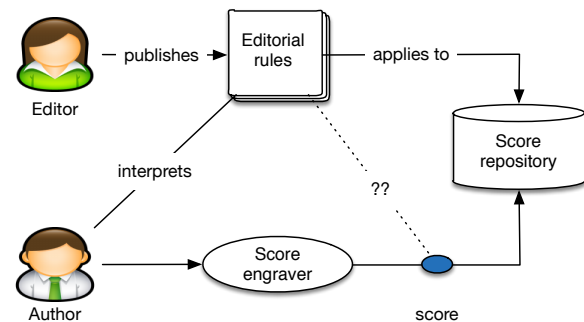


Figure 1. Editorial rules

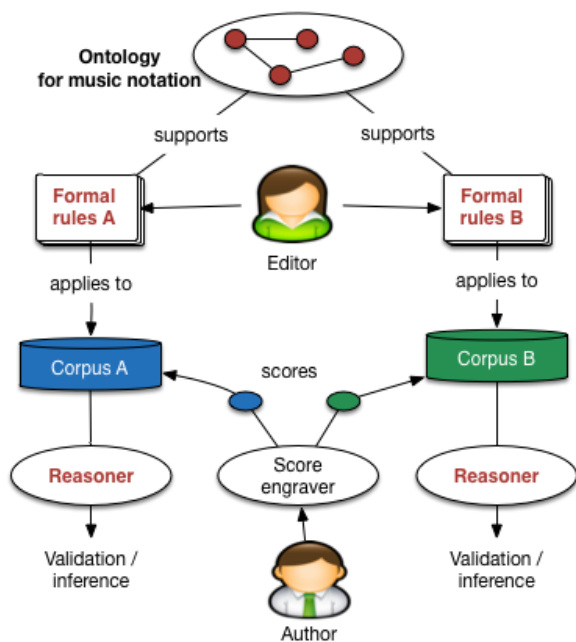
The approach is not fully satisfying. As shown by Fig. 1, there is no direct nor formal association between the rules and the encoded scores. The link depends on both the interpretation of a user, and the specific features of the score engraver. Even though we assume that the scores are edited by expert authors, keen to comply with the recommendations, nothing guarantees that they are not misinterpreted, or that the guidelines indeed result in a satisfying encoding. Moreover, rules that are not backed up by automatic validation safeguards are clearly non applicable in a collaborative context where un-controlled users are invited to contribute to the collections.

### 1.2 Formalizing editorial rules

Editorial rules are based on two important assumptions. First, they assume that both the editor and the author share a common *expertise* on music notation, and that this expertise supports rules, conveyed by sentences in natural language whose meaning is expected to be unambiguous.

Second, the author is assumed to “process” the rules while creating a new score, and guarantees that the resulting encoding fulfills them.

In the present paper we propose to formalize these assumptions, in such a way that *expertise*, *rules*, and *rules fulfillment* can all be explicitly stated and automatically validated. The main components of the approach are summarized by Fig. 2. Its foundations consists of an ontology of music notation, representing the concepts and domain-specific knowledge. Rules can be expressed (by an editor, possibly helped by experts) as formal sentences built from these concepts, and validation can be carried out by a reasoner that, given an instance of a score (interpreted as an instance of the ontology concepts), checks the rules fulfillment.



**Figure 2.** Formalization of rules

The rules might differ from one corpus to another, e.g., there is no reason to assume that the same set of constraints hold for a corpus of Renaissance music and for the Complete Works of Anton Webern. This approach therefore allows to specialize the definition of what a correct engraving is in a specific context, and can be seen as a complement of Finale, Sibelius or MuseScore that deliberately aim at proposing full-featured, non specialized engraving options.

The rest of the paper intends to demonstrate the promising perspective brought by associating a sophisticated encoding of music notation (say, using the MEI format) with knowledge-based management tools. We use as a driving motivation the expression and control of editorial rules on music scores corpora. Section 2 gives an overview of the approach, along with background notions. Section 3 proposes a simple ontology, and Section 4 examples of editorial rules. We discuss how the methodology can be used in a broader perspective in the concluding section.

## 2. BACKGROUND AND MOTIVATION

Let us first develop why, in our opinion, current technology falls short to support quality assessments on score encoding. We then provide some background on the field of formal ontologies and reasoning, and explain how this field can be used in the context of music notation.

### 2.1 Dealing with quality issues

The flexibility of music notation is such that it is difficult to express and check quality constraints on the representation that would universally hold. For instance, many formats we are aware of do not impose that the sequence of notes/rests in a measure exactly covers the measure duration defined by the time signature. As another example, in polyphonic music, nothing guarantees that the parts share common signatures and durations. So, even with the most sophisticated encoding, we may obtain a score presentation that does not correspond to a meaningful content (the definition of which is context-dependent), and will lead to an incorrect layout (if not a crash) with one of the possible renderers.

Besides, scores are being produced by individuals and institutions with highly variables motivations and skills. By “motivation”, we denote here the purpose of creating and editing a score in a digital format. A first one is obviously the production of material for performers, with various levels of demands. Some users may content themselves with schematic notation of simple songs, whereas others will aim at professional editing with high quality standards. The focus here is on rendering, readability and manageability of the score sheets in performance situation. Another category of users (with, probably, some overlap) are scientific editors, whose purpose is rather an accurate and long-term preservation of the source content (including variants and composer’s annotations). The focus will be put on completeness: all variants are represented, editor’s corrections are fully documented, links are provided to other resources if relevant, and collections are constrained by carefully crafted editorial rules. Overall, the quality of such projects is estimated by the ability of a document to convey as respectfully as possible the composer’s intent as it can be perceived through the available sources. Librarians are particularly interested by the searchability of their collections, with rich annotations linked to taxonomies [5]. We finally mention analysts, teachers and musicologists: their focus is put on the core music material, minoring rendering concerns. In such a context, part of the content may be missing without harm; accuracy, accessibility and clarity of the features investigated by the analytic process are the main quality factors.

Finally, even with modern editors, qualified authors, and strong guidelines, mistakes are unavoidable. Editing music is a creative process, sometimes akin to a free drawing of some graphic features whose interpretation is beyond the software constraint checking capacities. A same result may also be achieved with different options (e.g., the layer feature of Finale), sometimes yielding a weird and convoluted encoding, with unpredictable rendering when submitted to another renderer.

## 2.2 Knowledge formalization with OWL ontologies

One of the major achievement of the Semantic Web initiative [6] is the development of OWL, a language to represent *ontologies*.<sup>1</sup> An ontology is a set of axioms and rules that provide formal statements about the concepts (or “classes”) and concept occurrences (or “individuals”) of some knowledge domain. For instance, *Note* is a basic concept, which can be represented by a class in an OWL ontology, and some A4 in a score is an occurrence of the concept which can as well be represented in the ontology as an individual.

OWL supports inference mechanisms that derive new facts from those explicitly present in the ontology. As a trivial example, since A4 is a *Note*, which itself is a sub-class of *Sound*, a reasoner can infer that A4 is a *Sound*.

*Ontologies* have been recognized as an essential component for representing knowledge. An ontology commonly agreed to in a given domain constitute an essential basis to express formal statement that represent some domain knowledge, and to build sound reasoning and inference mechanisms related to this knowledge. The formalization of ontologies and reasoning also allows to automatically and safely validate facts, rules and constraints. As such, it constitutes an invaluable support to make sense to massive amounts of semi-structured data that would otherwise be hardly interpretable. While the initial purpose of semantic web technology is the mastering of Web data, its use has now spread to highly specialized knowledge domains. We make the case here for applying this approach to music notation.

The ontology proposed here is formalized using the fragments of OWL 2 [7] corresponding to the description logic  $SRQIQ(D)$  [8]. The use of OWL 2 is privileged because it provides a high expressiveness allowing semantic reasoners to verify the consistency of data, to derive new knowledge or to extract information already present. In addition, rules can be added to the ontology to express complex knowledge and provide more inference possibilities. A language of choice is SWRL, the Semantic Web Rule Language [9], which is briefly introduced in Section 4 along with rules examples.

## 2.3 Ontology-based quality assessment

Axioms and rules that compose an ontology can be used to assess the quality of music notation, assuming the latter is represented in some structured format (e.g., Kern, MusicXML, MEI, etc.). We can then interpret the content of a score in terms of the ontology concepts (see Fig. 3 for an illustration). This helps to reduce the conversion of notation elements from a score into *facts* representing concepts occurrences such as, e.g., “in this voice, in this measure, and for this duration, we find *this* chord.” The set of facts that we obtain together represent the notational knowledge encoded in the score, and we can then confront this knowledge to rules that state what are the fair facts.

In formal terms, new facts are produced and a reasoner can check if the ontology, augmented with facts and rules,

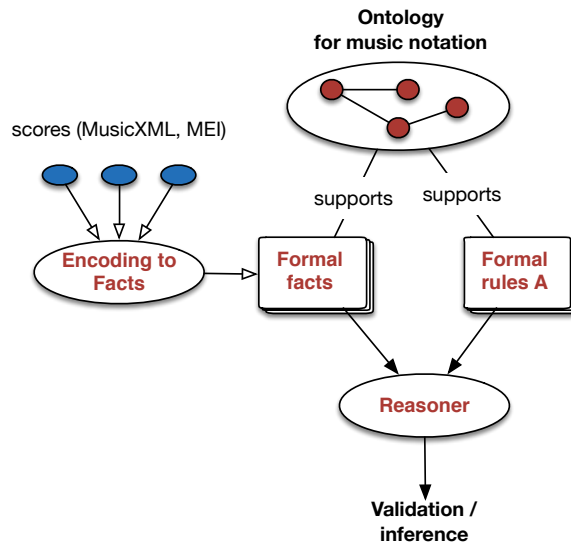


Figure 3. Representing a score as ontology facts

is still consistent and hence provide an information about the notation quality (e.g. accuracy, correctness, etc.). For instance, a fact which states that an event is at the same time a lyric and a rest, introduces an inconsistency when the ontology contains an axiom that says that an event is a disjoint union of these two classes. As another example, a rule stating that time intervals of two different events can’t overlap, helps to detect imprecisions in the expression of intervals related to voice events.

As illustrated by Fig. 3, the only non-standard component in this validation process (assuming a well-accepted ontology of music notation) is the converter that takes some score encoding as input and produces facts (usually encoded in RDF) as output. Implementating such conversion is definitely easy, and this makes the approach a quite attractive one, given its potential benefits.

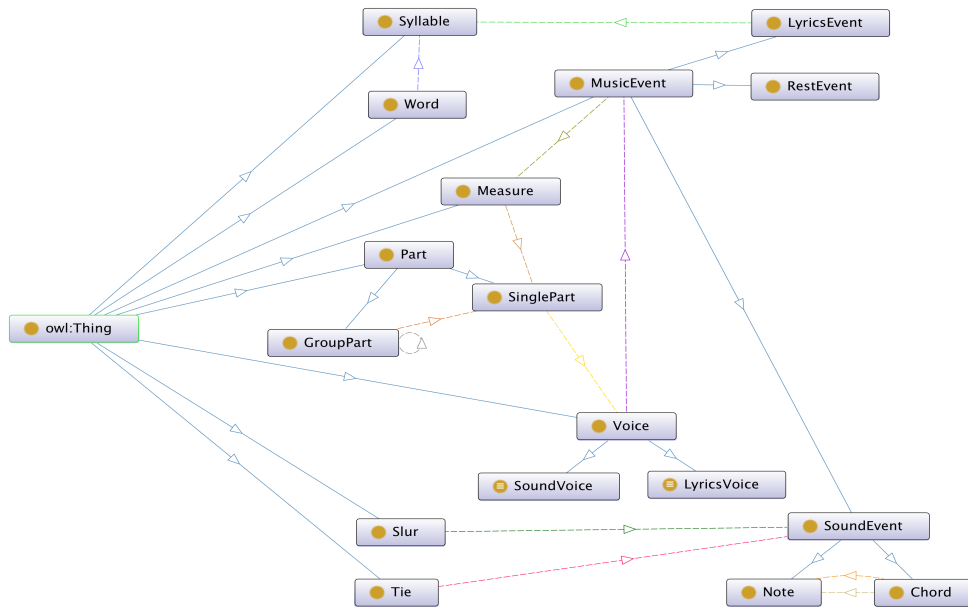
## 3. THE ONTOLOGY

We now present a concrete application of the above principles, based on an ontology of music notation specifically designed as a support for expressing quality constraints. Some preliminary words of caution are here in order.

### 3.1 Goals and Restrictions

The popularity and advantages of ontologies led to their usage in managing musical information. We can find high level or meta data oriented ontologies to manage metadata about musical works [10]. For more content oriented usage, Raimond and al. proposed the Music Ontology [11] to manage basic information about musical works and artists. The objective is to integrate musical works in the Semantic Web and the ontology is consequently used as a base for many music-oriented web services. A similar work is the Kanzaki Ontology [12], a music vocabulary which describes classical music and performances. These ontologies describes music at a work level and are suitable to describe general informations such as music categories

<sup>1</sup> <https://www.w3.org/TR/owl-features/>



**Figure 4.** The music notation ontology (OWL modeling)

(Chamber music, Choral music etc.) and performances (musicians, instruments etc.). To handle musical content descriptions, to enhance and facilitate their sharing among communities of both novices and experts, there is a need for more content oriented ontologies. We do believe that such a model would be of invaluable help to let the community formalize discussions and proposals and address issues related to the topic. We hope that the present work, although quite limited in its scope, can serve as an encouragement to initiate such an endeavour.

The proposed ontology is by no means intended to cover the whole knowledge of music notation throughout ages; this would be an extremely ambitious task (at least for the paper’s authors) which, at the very least, would require a long, collaborative process. The part of music notation that we aim at modeling here deliberately ignores issues related to the graphical layout of score. This aspect is major in estimating the quality of a score, as witnessed by the countless recommendations that can be found in reference such as [1]. However, it also constitutes a part of quality assessment which can hardly be evaluated from the encoding found in MusicXML or MEI formats. For the sake of simplicity and validation of our approach, we therefore chose to focus on the part of the notation that relates to “music content” in the following. Separating content from layout is not trivial, and to the best of our knowledge there does not exist a common agreement on this issue. We do not pretend to solve it here, but used the intuitive distinction between layout and content as a guideline to support the following decisions:

- All pure graphic instructions: paper size, margin, fonts, glyphs and positioning coordinates, are not considered.
- Directions regarding the assignment of voices and parts on staff are also ignored; this include the clef and textual annotations associated with staves.

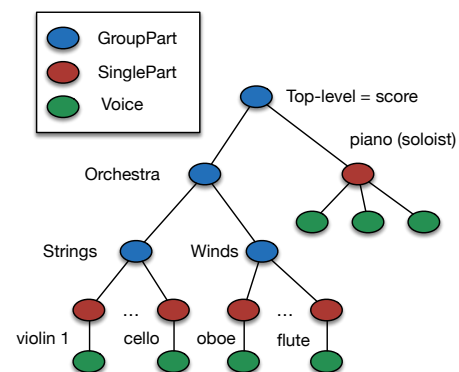
This essentially leaves elements that organize the music content are parts, parts in voices and voices as sequences of events. This is elaborated next.

### 3.2 The MusicNote Ontology

Fig. 4 shows the main concepts of our ontology<sup>2</sup>. The figure is produced by the *Protégé* editor<sup>3</sup>. The explanations that follow should make the major features clear event to non-experts. Essentially, a score is modeled as a *hierarchical structure*, where leaves consist of *voices*, and inner nodes of *parts*. A voice is a sequence of *events*, occurrences of the abstract class *Event* which is refined in several sub-classes. We detail first the structural aspect, then the voice representation.

#### 3.2.1 Structural aspects

Let us explain the structural aspect first by taking as an illustration the sketch of a piano concerto score (Fig. 5).



**Figure 5.** Structure of a score (GroupPart and SinglePart concepts)

<sup>2</sup> <http://cedric.cnam.fr/isid/ontologies/files/MusicNote.html>

<sup>3</sup> <http://protege.stanford.edu/>

A score is made of *parts*. Class `Part` represents an abstract concept which is refined in two sub-concepts:

1. `GroupPart`. A *group* (of parts) consists of a set of subparts, and mostly serves the organisational aspect of the score. For instance (Fig. 5), the orchestral material of a concerto score typically defines a group for wind instruments, another one for string instruments, etc.
2. `SinglePart`. A *single part* encapsulates the music notation elements assigned to an individual performer (instrument or vocal). Fig. 5 shows for instance a single part for the soloist (piano), another one for the violins, cellos, etc.

The content of group part may actually consist of an heterogeneous association of single and group parts, as illustrated by the top-level node of Fig. 5 that associates a group (the orchestra) and a single part (the soloist). This is reflected in the modeling of the ontology.

Note that we do not explicitly introduce a *score* concept. In our model, a score is simply the root of the tree of parts, and everything it contains. Such a concept would however be useful to introduce score-level metadata (composer, title, etc.) that would come as siblings of the parts hierarchy. Since we focus on the music content representation, we safely keep the model simple.

### 3.2.2 Core concepts: events and voices

A single part is a container for the core elements of music notation: events and voices. An *event* denotes the production of a noise artifact during a specific time period, called the *duration* of the event. Note that duration in this context has an absolute meaning, and corresponds to an open time interval fully contained in the temporal coverage of the score. The event concept can be refined based on the nature of the produced noise: it can be a sound (`SoundEvent`), text or syllables to be sung (`SyllEvent`) or a silence (`.`). The `SoundEvent` concept itself is decomposed as follows:

- `Note` denotes a simple, non-decomposable sound that can be represented by the well-known attributes `pitch`, `octave` and `accidental`. A more radical choice would be to simply represent a note by its frequency.
- `Chord` is an event composed of at least two notes that all share the same duration.

The status of `RestEvent` is debatable. A rest can be interpreted as an absence of event for a certain duration, and, in a radical perspective that would try to forget the idiosyncratic aspects of music notation, there is *a priori* no need to supply such a concept. One could also argue that rests are first-class notational objects that deserve to be explicitly represented. We can probably find contexts where a half rest is more appropriately represented as two quarter rests. A true, complete modeling of music ontology would have to carefully examine such cases in order to reach a large agreement.

A *voice* is a sequence of events whose durations do not overlap. A voice extends over a time range that can (optionally) be decomposed as a *sequence* of *measures*. A property of a measure is the *time signature*, the value of which can (extreme case) vary from one measure to another.

In summary, a score can essentially be seen, in our model, as a synchronization of an unbounded number of parts, each defining an internal organization of a finite time range split in measures.

### 3.2.3 A full example



Figure 6. A full example

Let us consider as a full example the score shown in Fig. 6, and its modeling. It consists of two parts, let's call them "vocal" and "accompaniment". The vocal part consists (in our modeling) of two voices, the first one (called "sopr") composed of sounds, and the second one ("lyrics") of syllables (note that there is no one-to-one rhythmic correspondence between syllables and notes, as some syllables cover several notes). The second part consists of a single voice, "bass". The structure is summarized by Fig 7.

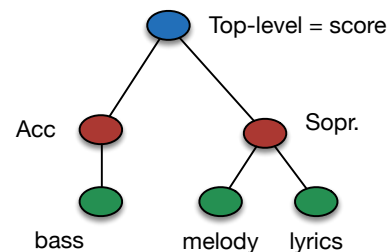


Figure 7. Structure of the example score

Consider now the details of each voice (summarized by Fig. 8). Voice "sopr" is a monophonic voice, instance of `SoundVoice`, each event being either a single note or a rest. Voice "lyrics", instance of `LyricsVoice`, consists of syllables. Finally, voice "bass", instance of `SoundVoice`, contains a few complex events, instance of `Chords`.

This example shows the main feature of how we can interpret a score notation as fact stated with respect to the ontology context. Those facts can automatically be extracted from the MusicXML or MEI encoding, represented in a convenient form (typically as RDF triples) and sent, along with ontology and rules, to a reasoner that will determine the consistency of the whole. Among other motivations, this can serve as a setting to validate quality rules, as discussed in the next section.

$$\begin{aligned}
v_{sopr}(t) &= \begin{cases} \perp, & t \in [0, 12[ \\ D5, & t \in [12, 20[ \\ \perp, & t \in [20, 22[ \\ E5, & t \in [22, 23[ \\ F5, & t \in [23, 24[ \\ D5, & t \in [24, 28[ \\ C\#5, & t \in [28, 32[ \\ \perp, & t \in [32, 34[ \\ A4, & t \in [34, 36[ \end{cases} \\
v_{lyrics}(t) &= \begin{cases} \perp & t \in [0, 12[ \\ Ah, & t \in [12, 20[ \\ \perp, & t \in [20, 22[ \\ que, & t \in [22, 23[ \\ je, & t \in [23, 24[ \\ sens, & t \in [24, 32[ \\ \perp, & t \in [32, 34[ \\ d'in, & t \in [34, 36[ \end{cases} \\
v_{bass}(t) &= \begin{cases} D4, & t \in [0, 8[ \\ C4, & t \in [8, 12[ \\ \langle B3es, D4 \rangle, & t \in [12, 16[ \\ A3, & t \in [16, 20[ \\ G3, & t \in [20, 24[ \\ \langle A3, C4is \rangle, & t \in [24, 30[ \\ G3, & t \in [30, 32[ \\ F3, & t \in [32, 36[ \end{cases}
\end{aligned}$$

**Figure 8.** Voices as sequences of events (measures 1 to 3)

#### 4. QUALITY RULES

Rules express constraints that music scores should respect. A language of choice to express rules is SWRL [9] which is briefly introduced first. We then enumerate some of the quality rules that can be expressed in this OWL+SWRL framework and conclude the section with few examples.

##### 4.1 SWRL

SWRL is a language that allows to express rules that take the form of an implication: a *body* is a list of statements which are interpreted as true or false depending on the context, and *head* is a statement whose truth value is inferred from the evaluation of the body. It is basically similar to rules in Datalog [13]. Let's take a simple example: the following rules define a `MajorChord` as a `Chord` with three notes `a`, `b`, `c` such that the interval between `a` and `b` is a major third (4 semi-tones) and the interval between `b` and `c` is a minor third (3 semi-tones)<sup>4</sup>.

```

Chord(?x), hasNote(?x, ?a), hasNote(?x, ?b),
           hasNote(?x, ?c),
           gap(?a, ?b, 4), gap(?b, ?c, 3)
  -> MajorChord(?x)

```

Symbols of the form `?x` denote variables. The interpretation of this rule is essentially: if we can find an instantiation of the variables `x`, `a`, `b` and `c` such that the body of the rule is evaluated as true, then we can infer that the head is true as well, i.e., `x` is an occurrence of the new, intentional concept `MajorChord`.

This simple example shows the kind of reasoning that allows to produce new knowledge about a set of facts (taken from a score encoding), and given a modeling of the domain supplied by a generic ontology. The `MajorChord` can now be reused just as any other concept, and we can thus build sophisticated reasoning chains that can be evaluated by a reasoner on a score or a corpus of scores. Let us examine the application of this idea for quality assessment.

##### 4.2 Quality dimensions

Quality measures are commonly organized according to the following quality dimensions [14]: *accuracy*, *completeness*, *trust* and *consistency*. We give below, for each dimension, some possible examples of quality rules for music notation.

<sup>4</sup> For the sake of illustration, the example rely on obvious simplifying assumptions.

*Accuracy* measures in what extend data values correspond to their considered correct representation. Classically, two kinds of accuracy are considered: the syntactic accuracy and the semantic one. The syntactic accuracy measures the adequacy of data to its expected format. A typical syntactic accuracy rule could check that (AccR1) each note is an existing one (roughly speaking in the domain  $\{C, D, E, F, G, A, B, C\}$ ), or that (AccR2) a voice nomenclature is respected, for instance with voices in the domain  $\{\text{Superius; Cantus; Altus; Contratenor}\}$ , or that (AccR3) at most one syllable is associated with a note.

The semantic accuracy measures the closeness of a value to a considered true real-world value. Its measurement supposes that there is somewhere a reference for the content to be checked, namely a business expert knowledge or another source to compare to. A syntactic accuracy rule could check that (AccR4) the birthdate associated with each compositor corresponds to the birthdate of a trusted other internal or external given source (e.g. Wikipedia if considered as trustable enough).

The *Completeness* measures in what extent the score contains all the required information, concerning data and meta-data. A syntactic completeness rule could check that (CompR1) a figured bass is present, or that (CompR2) at least one syllable is associated with each note, or that (CompR3) each measure is complete according to the figured bass, or (CompR4) the presence of some meta-data.

The *Trust* dimension concerns the trust-worthiness of each dataset, for instance by (TrustR1) checking the provenance information and the confidence in the provider.

The *Consistency* measures the adequacy of data to semantic rules. Such semantic rules may concern any element of the music score. A consistency rule could check that (ConstR1) each note can be played by the instrument (or voice) it is associated with, or that (ConstR2) the musical instruments were created before the compositor date of death.

##### 4.3 Rules expression

Rules such as those above can be expressed with SWRL according to the ontology defined in Section 3. For instance, the rule (AccR1) may be expressed by the following formula.

$$\text{Note} \equiv \{A\} \sqcup \{B\} \sqcup \{C\} \sqcup \{D\} \sqcup \{E\} \sqcup \{F\} \sqcup \{G\} \quad (1)$$

As another example, the following rule specializes (ConstR1) by stating that if a measure (?m) includes several notes (?e1) and (?e2) played at the same time (?i1 =?i2) then it belongs to a part (?pt) to which is associated a polyphonic instrument (?inst).

$$\begin{aligned}
 & Part(?pt) \wedge hasInstrument(?pt, ?inst) \\
 & \quad \wedge Measure(?m) \wedge hasPart(?m, ?pt) \\
 & \quad \wedge SoundEvent(?e1) \wedge SoundEvent(?e2) \\
 & \quad \quad \wedge differentFrom(?e1, ?e2) \quad (2) \\
 & \quad \quad \wedge hasMeasure(?e1, ?m) \wedge hasMeasure(?e2, ?m) \\
 & \quad \quad \wedge during(?e1, ?i1) \wedge during(?e2, ?i2) \wedge equals(?i1, ?i2) \\
 & \quad \quad \quad \implies Polyphonic(?inst)
 \end{aligned}$$

As shown by the complexity of this last rule, this requires either a close cooperation between a domain expert (e.g., a musicologist, a librarian) and a OWL/SWRL expert, or advances interfaces that let users build their own rules and control their meaning. This constitutes therefore both an exciting and promising axis for interdisciplinary research.

## 5. CONCLUSION

We presented in this paper an approach that aims at manipulating the content of music notation at a high level of abstraction, using concepts, knowledge and rules that leverage traditional encoding formats. The proposed methodology relies on OWL / SWRL, and we outlined the main steps: formal domain modeling with an OWL ontology, production of facts from the content of MusicXML or MEI documents, expression of rules, and production of new facts and knowledge thanks to a reasoner.

The work presented here is in progress, and is intended both to demonstrate to the TENOR community what can potentially be achieved with techniques that, as far as we know, have not yet been investigated in the music notation domain, and to encourage feedback or direct participation. Building an ontology requires all kinds of expertise, and aims at reaching the largest possible agreement. The present proposal is a step in this direction.

We are currently implementing a platform that focuses on quality evaluation rules. This is motivated by practical needs (we maintain on-line cooperative corpus for which quality issues are a primary concern). This restriction also makes investigations and experiments easier. We expect to be able to demonstrate the platform features during the conference, and hope that it will encourage discussions with the TENOR participants beyond notation quality issues.

## Acknowledgments

This work is partially supported by the GioQoSo project<sup>5</sup>, and the ANR MuNIR project<sup>6</sup>.

## 6. REFERENCES

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

<sup>5</sup> <http://gioqoso.irisa.fr/>

<sup>6</sup> <http://cedric.cnam.fr/index.php/labo/projet/view?id=41>

- [2] M. Good, *MusicXML for Notation and Analysis*. W. B. Hewlett and E. Selfridge-Field, MIT Press, 2001, pp. 113–124.
- [3] P. Rolland, “The Music Encoding Initiative (MEI),” in *Proc. Intl. Conf. on Musical Applications Using XML*, 2002, pp. 55–59.
- [4] “Music Encoding Initiative,” <http://music-encoding.org>, Music Encoding Initiative, 2015, accessed Oct. 2015.
- [5] J. Riley and C. A. Mayer, “Ask a Librarian: The Role of Librarians in the Music Information Retrieval,” in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2006.
- [6] T. Berners-Lee, J. Hendler, and O. Lassila, “The Semantic Web. A New Form of Web Content that is Meaningful to Computers will Unleash a Revolution of New Possibilities,” *Scientific American*, vol. 284, no. 5, pp. 34–43, May 2001.
- [7] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph, “OWL 2 Web Ontology Language Primer (Second Edition),” World Wide Web Consortium, W3C Recommendation, December 2012. [Online]. Available: <http://www.w3.org/TR/owl2-primer/>
- [8] I. Horrocks, O. Kutz, and U. Sattler, “The Even More Irresistible SROIQ,” in *KR*, 2006, pp. 57–67.
- [9] I. Horrocks and P. F. Patel-Schneider, “A proposal for an owl rules language,” in *In Proc. of the Thirteenth International World Wide Web Conference (WWW 2004)*. ACM, 2004, pp. 723–731.
- [10] L. Dodds. (2004) Musicbrainz metadata vocabulary. [Online]. Available: <http://www.ldodds.com/projects/musicbrainz/schema/mb.html>
- [11] Y. Raimond, S. Abdallah, M. Sandler, and F. Giasson, “the music ontology,” in *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2007, pp. 417–422.
- [12] M. Kanzaki. (2003) Music vocabulary. [Online]. Available: <http://www.kanzaki.com/ns/music>
- [13] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*. Addison-Wesley, 1995.
- [14] C. Batini and M. Scannapieco, *Data Quality: Concepts, Methodologies and Techniques*, ser. Data-Centric Systems and Applications. Springer, 2016.