Towards the Future of Multi-Layer Music Encoding: The IEEE 1599 v2.0 Draft

Adriano Baratè, Goffredo Haus, Luca A. Ludovico
Laboratory of Music Informatics (LIM)
Department of Computer Science
University of Milan, Italy
{first.last}@unimi.it

Davide A. Mauro
Department of Computer Sciences
and Electrical Engineering
Marshall University, USA
maurod@marshall.edu

***Abstract*—In this contribution, we present the latest proposed changes to the IEEE 1599 format. IEEE 1599 is an XML-based format first standardized in 2008 by the Computer Society of the Institute of Electrical and Electronics Engineers (IEEE CS). The original goal of the standard was to provide a digital format capable of describing music pieces, supporting the presence of heterogeneous materials (symbolic, formal, graphical, audio, etc.), inside an XML container. The format allowed advanced features such as a multi-layer information description and synchronization among entities (known as layers). Since its conception, the technological landscape has changed and new and different needs have emerged. The experience gained over the years has highlighted the strengths and limitations of the format and, for these reasons, the *IEEE Working Group for XML Musical Application* (IEEE WG_1599) is now contributing to a new release of the format. This paper will use a commented example to illustrate the proposed changes together with their impact compared to the original standard.**

1. Introduction

IEEE 1599 is an international standard for the representation of music information in XML. It is a multilayered environment integrating general (metadata), structural, notational, computer-driven performance, and audio layers. In the original intentions of its creators, such a structure should allow for a comprehensive representation of music information, from basic score notation to detailed instructions for a computer to perform the music.

IEEE 1599 was conceived as a comprehensive format for the representation and manipulation of music information, usable for a variety of purposes, including the creation and editing of music scores, the storage and retrieval of music information, the computer-based performance of music, different kinds of musical and musicological analyses, the education of music students, and the advanced experience of intangible cultural heritage.

Officially standardized in 2008 by the *Institute of Electrical and Electronics Engineers Standards Association* (IEEE SA),[1] IEEE 1599 has been studied under different perspectives in dozens of scientific papers and applied in relevant international projects and products.[2] This standard opened up opportunities for novel applications in music experience, publishing, and research, including innovative multimedia products, music-oriented educational platforms, and software tools for preserving and utilizing cultural heritage.

Ten years after IEEE 1599 approval, the research team that originally conceived the standard, namely the *IEEE Working Group for XML Musical Application* (IEEE WG_1599), suggested revising it. Several factors favored this decision. First, the technological landscape has changed drastically during the last decades. To mention but a few novelties, in the period after the standardization of IEEE 1599, we have experienced the availability of new network technologies such as 5G to deliver multimedia streams, significant advancements in the automatic extraction of features from audio and symbolic musical documents, and the flourishing of generative artificial intelligence and its impact on music composition.

At the same time, other formats with strong relevance for IEEE 1599 (e.g., MEI and MusicXML) have been developed. Finally, the usage of the standard over this time, including the development of authoring tools and the implementation of interactive viewers, allowed the working group to gain new insights into the strengths and shortcomings of the format.

Although the core and focus of IEEE 1599 remain the same, several technical and foundational approaches can be updated to better reflect current technologies, the desire to facilitate the extensibility of the format, and new use cases [1]. In this sense, a case study under-analyzed in the first version of the format is represented by machine-driven symbolic music generation [2]. Moreover, compared to the release period of IEEE 1599, nowadays the diffusion of the Semantic Web has increased significantly [3], with a potential impact also on the way of describing musical information on the network. Finally, a recent groundbreaking scenario is the one provided by the Internet of Musical

1. https://standards.ieee.org/ieee/1599/4403/

2. For an up-to-date list of publications, projects, and products dealing with IEEE 1599, please refer to the *Documentation* and *In practice* areas of the official web site at https://ieee1599.lim.di.unimi.it/.

Things [4], where exchange formats are an integral part of the vision and standardization activities are largely unrealized.

This contribution aims to illustrate the proposed changes to the definition of the IEEE 1599 standard that are currently being examined within the IEEE working group. After discussions with the scientific community, stakeholders, and industry experts, the refinement of these ideas is expected to drive the writing of the IEEE 1599 v2.0 draft.

The rest of the paper is structured as follows: Section 2 will provide a short overview of IEEE 1599 and present the milestones of the project so far, Section 3 will highlight the most notable shortcomings of the format, Section 4 will introduce the changes that should characterize the new version of the standard, together with a minimal example of an IEEE 1599 v2.0 document, Section 5 will comment on a clarifying case study, and, finally, Section 6 will draw the conclusions and pave the way for future work.

2. Overview and History of IEEE 1599

2.1. Key Features of the Standard

In the last decades, new interactive music services have arisen, often employing proprietary file formats. The adoption of a standardized file format could enhance interoperability among these services. For instance, in pursuit of this goal, the ISO/IEC Moving Picture Experts Group (MPEG) in 2010 published a standard named ISO/IEC 23000-12:2010 and known as *Multimedia application format (MPEG-A) – Interactive Music Application Format* (IM AF).[3] Differently from IEEE 1599, the main focus of this format was on multimedia [5].

IEEE 1599, standardized by IEEE SA in 2008, serves as a comprehensive format for describing music. Based on XML, it offers a multi-layered environment that allows for the integration of various aspects of music information, including general metadata, structural elements, notation, computer-driven performance data, and audio/video tracks. Typically, music information encoding involves the adoption of distinct reference formats for audio (e.g., CD-DA, DVD-A, FLAC, MP3, AAC), computer-driven performance (e.g., MIDI, MPEG, SASL/SAOL), symbolic music (e.g., Plaine & Easie Code, MEI, MusicXML), and scores (e.g., JPEG, PNG, TIFF). Each of these formats focuses on specific aspects of musical information, leaving gaps in comprehensive representation. The need for integration arises to provide interactive and synchronized access to all layers of music (audio, performance, music notation, musical forms, and metadata). Such integration facilitates activities like following a music score while listening to the corresponding audio, real-time comparison of graphical representations and audio performances, and interaction in a multi-modal environment with musical content. By harmonizing music representation with existing accepted standards and formats, IEEE 1599 facilitates seamless interchangeability among different applications and acts as a hub for diverse music-related content.

This standard caters to a wide range of software applications dealing with music information, such as digital score editors, optical music recognition (OMR) systems, web and mobile apps, musical databases and archives, and performance, composition, and musicology-oriented tools. Incorporating all music-related information into XML, IEEE 1599 benefits from being hierarchical, extensible, portable, and both machine and human-readable. Moreover, IEEE 1599 is a flexible standard that can represent a wide variety of music genres and styles; it is an open standard, which means that it is freely available to use and distribute; it is well documented and easy to learn and use.

A detailed description of the characteristics of the standard would be beyond the scope of this paper and would reproduce the contents already available in other publications [6]. For further details, the reader is invited to visit the official website,[4] which offers many resources focusing on the original version of the standard, including musical examples and official documentation.

2.2. Milestones of the Project

Building on top of ideas dating back to the late 1970s, the official history of the IEEE 1599 standard now spans four decades. As mentioned earlier, the initiatives related to the format standardization have taken place under the umbrella of the Institute of Electrical and Electronics Engineers (IEEE), and in particular the IEEE Computer Society (IEEE CS) and the IEEE Standards Association (IEEE SA). In this section, we present the main steps of the project:

- In 1992, the IEEE CS *Task Force on Computer Generated Music* was created, followed by the establishment of the IEEE *Technical Committee on Computer Generated Music* (IEEE TCCGM) in 1994;
- In 2001, the IEEE SA *Working Group on Music Application of XML* was created, and the approval of PAR1599, a recommended practice for the "Definition of a Commonly Acceptable Musical Application Using the XML Language," was granted by IEEE SA;
- The *IEEE International Conference on Musical Application using XML* (MAX 2002) took place at the University of Milan in 2002, where the future standard was officially presented to the scientific community under the name of MX;
- In 2008, the IEEE balloting phase ended, resulting in IEEE 1599 becoming an international standard;
- By 2011, *Project EMIPIU* (standing for Enhanced Music Interactive Platform for Internet User) released a web player for IEEE 1599, which still represents the engine for IEEE 1599 browser applications;
- In 2013, the book "Music Navigation with Symbols and Layers: Toward Content Browsing with IEEE 1599 XML Encoding," edited by Denis L. Baggi and Gof-

3. https://www.iso.org/standard/53644.html

4. https://ieee1599.lim.di.unimi.it/

fredo Haus, was published by Wiley-IEEE Computer Society [6];
- The IEEE Standards Association approved the revision draft for the standard in 2018;
- In 2019, the *1st International Workshop on Multilayer Music Representation and Processing* (MMRP19) was held at the University of Milan,[5] also hosting the kickoff meeting of the IEEE WG_1599 for the revision of the standard. At the moment of writing, the second edition of the workshop, known as MMRP23, is in progress at the University of Pisa.[6]

During the early 2000s, a number of other initiatives looked at XML as a tool for encoding music information. In particular it is worth mentioning the *WEDELMUSIC* format (see [7]) that appears to be no longer maintained, and the seminal work of M. Good regarding MusicXML (see [8] and [9]).

3. Shortcomings of the Format

Ten years after the standardization of IEEE 1599, some shortcomings in its use and applicability to information encoding have emerged. Please note that the standard has also been tested in domains far from music but somehow connected to the synchronized experience of multimedia objects. Examples include live theatrical performances [10], content and language integrated learning [11], and sightseeing tours [12]. In addition to direct experience, an inspiring paper dating back to 2016 offered a critical review of the IEEE 1599 standard [13]. That seminal work highlighted some relevant shortcomings related to the characteristics of the format. One of the goals of the current revision effort is to solve the issues that have emerged so far.

3.1. XML Schema vs. Document Type Definition

IEEE 1599, designed in the early 2000s, was defined through a Document Type Definition (DTD), like most XML-based formats of the time. Other relevant examples in the field of music encoding included the MEI format and MusicXML. XML Schema, originally published as a W3C recommendation in May 2001, was taking its first steps but it was still viewed with suspicion: “Anything that can be expressed with a DTD can be expressed with what is now called an XML schema. Eventually, all DTDs may be replaced by schemas, but applications based on DTDs, which are upwardly compatible with schemas, are in no danger of being made obsolete by schemas” [14].

Nowadays, both XML Schema and DTDs are commonly used to define the structure and rules for validating XML documents, but they differ in several key aspects that favor the former approach.

Concerning the syntax, XML Schema itself is written in XML format, which means that it uses XML elements and attributes to define the rules and constraints for the XML document. DTDs, on the other hand, have their own specific syntax, which is different from XML, using a combination of angle brackets and parentheses to define elements and their content models.

Moreover, XML Schema offers a more comprehensive set of features for defining complex data structures and data types. It allows defining data types such as string, integer, and date, and supports features like namespaces, inheritance, and more advanced data validations. DTDs, in contrast, are simpler and more limited in their capabilities. They lack support for complex data types and are more suitable for basic structural validation.

XML Schema has built-in support for XML namespaces, allowing you to define elements and attributes within specific scopes. DTDs do not have such a feature, which can make it challenging to define complex XML documents.

Concerning extensibility, XML Schema is extensible, which means that you can modify and extend existing schemas without having to rewrite the whole schema. DTDs are less extensible, and any changes or extensions may require more significant modifications to the DTD itself.

Finally, regarding compatibility and adoption, XML Schema is the more modern and widely adopted approach for defining XML document structures. It is the recommended choice (also by W3C) when working with newer XML technologies; while DTDs were widely used in the past and still work with many XML parsers, they are considered somewhat outdated compared to XML Schema.

Nowadays, XML Schema is generally preferred over DTD for most modern XML projects. XML Schema offers more flexibility, expressive power, and support for namespaces, making it better suited for complex data structures and more sophisticated validation requirements. Additionally, XML Schema has become the industry standard for defining XML document structures, and it is well supported by various XML processing libraries and tools. When redefining an XML-based format, it is necessary to consider all these aspects but also pay attention to backward compatibility.

3.2. Generalization of Layers

Probably the most impactful issue we want to address emerges from a paradigm shift in the vision of music description by the working group. Even if the scientific literature has historically identified six (or even fewer) layers capable of providing a comprehensive description of a musical piece (see, e.g., [15], [16], [17]), the prevailing idea is now to overcome this rigid structure and let users define their own description layers.

On the one hand, the format will be able to adapt to future changes and support representation domains that are currently unpredictable. On the other hand, descriptions that theoretically cover multiple layers will be better managed. For example, a *Max*[7] or a *Pure Data*[8] patch could be

5. https://mmrp19.di.unimi.it/
6. https://mmrp23.lim.di.unimi.it/
7. https://cycling74.com/products/max
8. https://puredata.info/

simultaneously considered as an instance of the *Logic* layer (it describes symbolic events), the *Structural* layer (it expresses relationships between musical objects), the *Notational* layer (it presents a graphical layout, even if different from Common Western Notation), and the *Performance* layer (it originates a computer-driven performance). Thanks to the new possibilities offered by IEEE 1599 v2.0, users will be able to create new layers, including one specifically dedicated to *Max* and *Pure Data* patches.

A consequence of the new vision is the need to detach the *spine*, namely the data structure that keeps the various layers connected in IEEE 1599 v1.0, from its current location, which is the *Logic* layer. In this way, the *Logic* layer will no longer play a privileged role in IEEE 1599, simply becoming one of many possible forms of descriptions for a music piece. For instance, compositions that cannot be described either in terms of standard notation or as a predefined sequence of events will benefit from this generalization.

As a desirable side effect, all layers will be similar in their structure and the resulting definition of the format will be simpler.

3.3. Score Formats in the *Logic* Layer

The *Logic* layer, i.e., the one devoted to the description of scores in terms of musical symbols, was originally conceived to support only a custom XML representation, standardized by the IEEE 1599 format itself. Conversely, other forms of description for music events (e.g., notational and audio descriptions) are open to already existing formats and, rather, oriented to identify, through an XML syntax, the position of the music event in external digital objects. In practical terms, an IEEE 1599 document can have, say, n external image files and m external audio files attached, but a single internal XML block to describe music symbols.

In the authors' opinion, this imposition clashes with the general vision of the format from two points of view: 1. it treats the *Logic* layer in a way different from other layers, somehow promoting the internal musical description as the reference one and causing asymmetry, and 2. hampering the adoption of already in-use alternative formats, such as MEI and MusicXML, to describe logical information.

3.4. Containers for Multiple Instances

For each IEEE 1599 layer (except the *Logic* one), a multiplicity of instances is supported. This means that the encoding of a music piece can host, say, n external image files linked to the *Notational* layer and m external audio files attached to the *Audio* layer.

Unfortunately, in the original format, another aspect of asymmetry was inadvertently introduced. In the *Notational* layer, there is an intermediate container structure that allows the organization of image files into collections, corresponding to score versions or editions. The meaning is clear since it is natural that a digital score is generally made up of a given number of scans.

A similar approach should also have been implemented for audio and video files. In fact, the lack of a container for multiple tracks in the *Audio* layer causes ambiguity in their meaning: Do they come from different performances of the same piece (i.e., are they alternative?) or should they form together a unique performance where single tracks are available for single parts (i.e., are they complementary?).

3.5. Digital Rights Management (DRM)

For a strongly multimedia-oriented format like IEEE 1599, where various media types and digital objects combine to create a rich description, accurate management of intellectual property and digital rights is a key aspect. One of the shortcomings that have been hampering the widespread diffusion of IEEE 1599 so far is the loose management of DRM, an aspect that discouraged stakeholders from investing in the format.

The problem was not completely ignored during the standardization process. In fact, IEEE 1599 specifications introduced generic elements to express licenses on each linked digital object. Nevertheless, in the absence of hardware and software architectures devoted to license validation, like those proposed in [18], it is easy to fraudulently manipulate the XML document and gain full access to protected materials.

4. Proposed Changes for v2.0

Following the outline discussed in Section 3, we want to illustrate in more detail the various changes and their impact on the format. Fig. 3 presents a minimal working example showing the new structure of the *spine*, the support for multiple *spine* instances, and the definition of new layers.

4.1. XSD Schema

The first and most foundational step is the implementation of the new version of the standard using XML Schema Definition (XSD) rather than a DTD. As explained in Section 3.1, XML Schema presents many advantages over DTDs: being written as an XML document, supporting namespaces, defining complex datatypes for elements and attributes, etc. There are existing software solutions that can help in the process of migration from DTD to XML Schema. For example, *Microsoft Visual Studio 2022*,[9] *IntelliJ IDEA Ultimate*,[10] and *Altova XMLSpy 2023*[11] offer tools to convert a DTD into XSD format. The result of such an automatic approach was manually validated and then used as a starting point to incorporate the new changes listed below.

Another option could be the migration toward a completely different way to encode information while maintaining the characteristics of interoperability, hierarchical structure, and readability. In this sense, JavaScript Object

9. https://visualstudio.microsoft.com/vs/
10. https://www.jetbrains.com/idea/
11. https://www.altova.com/xmlspy-xml-editor

Notation (JSON) could be a good choice. Even if the subject is under discussion in the working group, XML is still considered preferable since IEEE 1599 v2.0 is intended as an evolution of v1.0 and one of the goals of the initiative is to keep backward compatibility as much as possible.

4.2. Generalization of Layers

As mentioned in Section 3.2, IEEE 1599 v1.0 included six layers: *General*, *Logic* (including two sub-layers: *Spine* and *Logically Organized Symbols*), *Structural*, *Notational*, *Performance*, and *Audio*. The new version of the standard aims to generalize the concept of layer and delegate to each individual layer the definition of its own semantics in accordance with the following rules: 1) Each layer must have a unique identifier within the document; 2) Each layer is characterized by a name and a description; 3) A layer can refer to specific *spine* events with the *event_description* construct; 4) A layer can define its own semantics using an XML Schema or a namespace within an XML schema.

Indeed, the presence of a predefined set of nested elements within a fixed 6-layer structure, typical of the original version, has helped developers in releasing applications capable of fully supporting the syntax and correctly interpreting the semantics. Concerning IEEE 1599 v2.0, on the one hand, it homogenizes and intrinsically simplifies the syntax (see lines 37–61 and 62–64 in Fig. 3), but, on the other hand, it makes the semantics dependent on the user's intentions. In this sense, employing the encoded information in a meaningful way is left to ad hoc applications.

The conversion of IEEE 1599 v1.0 layers into v2.0 layers requires adapting them to the new definition without altering their semantics. For example, the v1.0 *Notational* layer can be transformed into a specialization of a generic v2.0 layer called "notational" (see lines 37–61 in Fig. 3).

4.3. Updates to the *Spine* Construct

In IEEE 1599 v2.0, we propose to introduce two main changes to the *spine*.

The first change is the removal of virtual timing units (VTU) and horizontal offsets for music events. In IEEE 1599 v1.0, VTUs were used to determine an implicit order within events according to an abstract "time axis". VTUs provided a way to linearize musical events that are typically placed on a 2-dimensional score, whereas XML requires a 1-dimensional order of elements. For example, marking right-hand events with `rh`, left-hand events with `lh` and arbitrarily choosing 8 VTUs for eighth notes, the *spine* of the excerpt shown in Fig. 1 would be:

```
<ieee1599 version="1.0">
  ...
  <logic>
    <spine>
      <event id="rh_01" timing="0" hpos="0" />
      <event id="lh_01" timing="0" hpos="0" />
      <event id="rh_02" timing="8" hpos="8" />
      <event id="lh_02" timing="0" hpos="12" />
      <event id="rh_03" timing="8" hpos="-4" />
      <event id="rh_04" timing="8" hpos="8" />
```

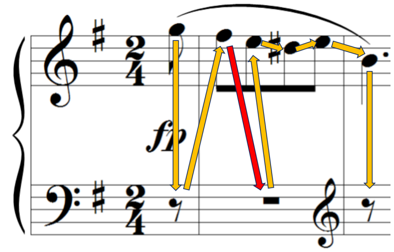

Figure 1. Arrows show the path used in the *spine* to linearize the musical symbols of a 2-dimensional score. The dark arrow highlights the only discrepancy between time (VTUs) and space (HPOS) values, due to a full-measure rest that virtually occurs at the beginning of the measure but is graphically shown in the middle.

```
      <event id="rh_05" timing="8" hpos="8" />
      <event id="rh_06" timing="8" hpos="8" />
      <event id="lh_03" timing="0" hpos="0" />
      ...
    </spine>
  </logic>
  ...
</ieee>
```

Similarly to the attribute `timing`, the attribute `hpos` allowed to express virtual spatial positions. Even if the virtual units for time and space could be completely independent, the vast majority of IEEE 1599 v1.0 examples kept values aligned. Please note that, in some cases, the best choice would have been to adopt different values, as shown by the dark arrow in Fig. 1 corresponding to events `lh_02` and `rh_03`. It is evident that this approach is related to a "traditional" representation of musical events. As shown, e.g., in lines 4–10 of Fig. 3, new *spine* events only contain unique identifiers that, in accordance with the original format, are intended to act as a glue among layers: when describing the same event, all layers will refer to the same value of the `id` attribute. There will be no `timing` and `hpos` attributes in the new version.

In the revision, each layer should be responsible for its own definition of the axes (if any) used to define the relative position of events. For example, it is natural to hypothesize the presence of a time axis for timed descriptions (e.g., the new version of the *Audio* layer) and space axes for graphical descriptions (e.g., the new version of the *Notational* layer).

The exact mechanism to express standardized positions in the layers, which are now completely user-defined, is still under study.

A possibility is to rely on already-defined ontologies. An example is provided by the *Ontology of units of Measure* (OM), which provides classes, instances, and properties that represent the different concepts used for defining and using measures and units.[12]

The second major change is the novel support offered for multiple *spines* within a single document. An IEEE 1599 v1.0 document was focused not only on a single musical piece but, implicitly, on a specific logical description of that piece. A certain version of the score was considered, rightly

12. http://www.ontology-of-units-of-measure.org/page/om-2

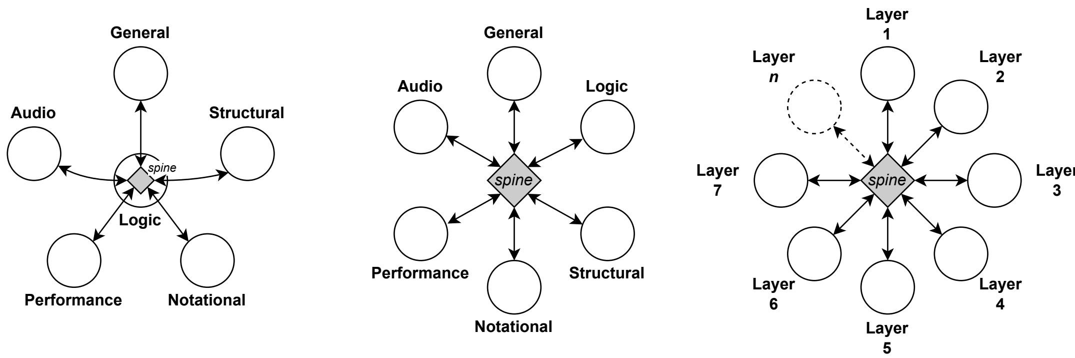

Figure 2. The evolution of layers and the *spine*. On the left is the original standard: the *spine* is a part of the *Logic* layer; in the middle is the evolution toward layer symmetry envisaged in [1]: the *spine* is the central node of a star, but layers are still predefined in their number and function; finally, on the right is the revision proposal, where n layers can be connected to the *spine*.

or wrongly, to be the author's last will and all variants had to relate to it. As a trivial example, a dotted quarter note, encoded as a single musical event in the *spine*, could be written in some score versions as an aggregation of symbols, e.g. a quarter note tied to an eighth note, and vice versa; mapping many notational symbols onto a single logical event or, conversely, one symbol onto many events could be confusing. As a more intriguing case, the availability of multiple spines lets a single XML document accommodate diverging versions of the same original composition (e.g., rearranged cover versions with different ensembles or music genres), which keep common events linked and synchronized. An example will be shown in Section 5. To this end, the format has to support not only the simultaneous presence of n *spine* structures (see lines 3–23 in Fig. 3), but also mappings between different *spines* (see lines 24–35 in Fig. 3).

4.4. Rethinking the *Logic* Layer

In the original standard, the presence of the *Logic* layer was required due to the need of hosting the *spine*. As mentioned above, nowadays, the idea is to move the *spine* outside of any layer. Consequently, in IEEE 1599 v2.0, there will be no predefined *Logic* layer. The possibility to describe music from a symbolic point of view will be clearly maintained, but this will occur through one of the specializations of generalized layers. The choice to include this layer and, in that case, how to describe musical symbols will be left to the author of the encoding. Already existing formats to encode symbolic music (e.g. MEI, MusicXML, Plaine & Easie Code) will be supported.

As a result, IEEE 1599 v2.0 should finally implement the perfect symmetry among layers proposed in [1], moreover generalized to n user-defined layers, as shown on the right side of Fig. 2.

4.5. Containers for Multiple Instances

Another issue mentioned in Section 3.4 concerns the availability of multiple-tier containers for instances. The generalization of the layer syntax will introduce an intermediate container structure capable of accommodating multiple layer-instance parts. Examples embrace the n scans (layer instance parts) that, together, constitute a single score (layer instance) or the m tracks (layer instance parts) that combine to form an audio performance (layer instance).

Such an approach solves the ambiguity mentioned above. Two different scores, as well as two different audio performances, will be described in separated layer-instance branches, which, in turn, will act as containers for all the digital objects involved in that specific description (see lines 38–59 in Fig. 3).

4.6. Backward Compatibility

Backward compatibility, intended here as the possibility to retrieve information from IEEE 1599 v1.0 documents, is a particularly important aspect of the update process. Over the years, a vast number of musical materials have been digitized, and many examples adhering to the original specifications have been produced. Consequently, it is a concern to ensure that those documents are still usable with the new software.

In order to maintain compatibility, two approaches are currently under study: i) the creation of ad hoc file converter utilities, and ii) the adoption of so-called *profiles*. The former way would be straightforward from the format perspective since v1.0 documents would be fully converted into v2.0 files, but it would require the development of suitable software tools. Conversely, the latter approach implies that some specialized layers with reserved names are internally encoded using the rules of IEEE 1599 v1.0. This methodology would be straightforward from the conversion perspective since whole XML blocks would be simply copied and pasted from old documents to specific locations in the new files, but it would require the formalization of a mechanism to identify and validate a mix of the old and the new syntax inside a single document.

In conclusion, the idea of integrating profiles seems to be very practical but syntactically inelegant. No decision has yet been made on this topic. In both scenarios, thanks to the readability typical of XML, it will be easy for human experts to compare the original and the converted files and check for possible issues.

5. Case Study

To better illustrate the proposed changes and their potential relevance in music description, we now present a case study focusing on the *Piano Concerto No.2, Op.18* – 2nd movement "Adagio sostenuto" by Sergei Rachmaninoff. For the sake of brevity, we will consider only a small excerpt: measures 13–17 of the clarinet part, graphically highlighted in yellow in Figure 4.

5.1. Multiple Logical Descriptions

The example refers to two text-based encoding formats: *Plaine & Easie Code* and the *Music Encoding Initiative* format. The first research question is how to link these alternative logical descriptions to the same *spine*.

Plaine & Easie Code (PEC) is a library standard that enables entering music incipits in modern or mensural notation. The code is maintained by the *International Association of Music Libraries, Archives and Documentation Centres* (IAML) and the *Répertoire International des Sources Musicales* (RISM) for use as an exchange format in the library environment.[13] The excerpt, including clef, key, and time signatures, is shown in Figure 6.

The *Music Encoding Initiative* (MEI) is a community-driven, open-source effort to define a system for encoding musical documents in a machine-readable structure.[14] The project started in 1999 and was first presented at ISMIR 2000 [19]. The initiative evolved and became one of the more well-established platforms for discussions regarding the subject of digital encoding of music. The excerpt is shown in Figure 7. The MEI initiative sponsors a yearly conference: Music Encoding Conference (MEC). Of particular relevance for our present work are the following contributions:

- Hankinson *et al.* [20] present how "MEI as a document-encoding framework [...] can be extended to encode new types of notation, eliminating the need for creating specialized and potentially incompatible notation encoding standards";
- Viglianti [21] informed our changes to the spine construct. The *Enhancing Music Notation Addressability* (EMA) project looked at methods for "addressing arbitrary portions of encoded music notation on the web";
- Parada-Cabaleiro and Torrente [22] and Goebl and Weigl [23] deal with the conversion issues across musical symbolic representations;
- Berndt [24] looks at how musical performances can be described in a systematic way and introduces a new XML format called *Music Performance Markup* (MPM).

13. https://www.iaml.info/plaine-easie-code
14. https://music-encoding.org/

Now we illustrate how to link the two mentioned descriptions to an IEEE 1599 v2.0 document. This example strictly requires a single *spine* containing 32 events, one per musical symbol (notes and rests). Both PEC and MEI representations fall into the category of logical descriptions; consequently, it is natural to include a single layer, called, e.g., "logic", containing two instances, both made of one layer part. Please note that, in the more complex case of a part-by-part description in MEI or PEC, the corresponding layer instance would host n layer parts. Finally, each layer part contains 32 event descriptions. Adhering to the theoretical framework of IEEE 1599, PEC and MEI syntax is kept outside the XML document, in the form of an external text file and XML file, respectively. The two files are linked to the main document thanks to the `src` attribute of the `layer_instance_part` element (see Fig. 3). Finally, concerning the identification of event positions in external files, different approaches can be adopted, depending on the characteristics of the formats. PEC is plain text, thus the position of events can be defined in terms of char offsets from the beginning of the file. Since some events span a number of characters, start and end positions must be expressed. MEI adheres to XML and `xml:id` attributes univocally identify XML blocks. In this case, IEEE 1599 can rely on the identifiers assigned by MEI, remapping them onto its own *spine* ids in order to relate them to other layer instances and layers.

5.2. Multiple *Spines*

The example under exam is made even more interesting by a crossover with pop music. Eric Carmen's 1975 hit, "All By Myself", brought to success by the interpretation of Celine Dion, clearly cited a theme from the 2nd movement. As proof of this fact, the song features both Eric Carmen and Sergei Rachmaninoff on its list of songwriters. One of the clearest citations is the one highlighted in Fig. 5; even if the key is different and there are variations in melody and rhythm, the melodic contour and the underlying chords are tightly connected to the clarinet part of the piano concerto.

This is the ideal scenario to apply one of the new features proposed for IEEE 1599 v2.0, namely the possibility to have multiple *spines*. Specifically, the encoding could include a *spine* made of 32 events for the clarinet part, another *spine* made of 46 events for the song's leading voice, and a number of cross-*spine* mappings to define correspondences between events.

The rest of the IEEE 1599 document could host descriptions and digital objects mainly referring to either the piano concerto or the pop song. For example, a *Video* layer could embrace video recordings of the piano concerto and TV footage of the pop song, linked to the first or the second *spine*'s events, respectively. In the common sections, *spine* mappings would logically allow jumping from the former to the latter and vice versa.

Needless to say, in order to exploit its potential, such a feature must be reflected by the new applications working with the format, in particular by media players.

```xml
<ieee1599 version="2.0">
   <spines>
      <spine id="spine_01" description="This is Spine #1">
         <event id="spine_01_ev_01" />
         <event id="spine_01_ev_02" />
         <event id="spine_01_ev_03" />
         <event id="spine_01_ev_04" />
         <event id="spine_01_ev_05" />
         <event id="spine_01_ev_06" />
         <event id="spine_01_ev_07" />
      </spine>
      <spine id="spine_02" description="This is Spine #2">
         <event id="spine_02_ev_01" />
         <event id="spine_02_ev_02" />
         <event id="spine_02_ev_03" />
         <event id="spine_02_ev_04" />
      </spine>
      <spine id="spine_03" description="This is Spine #3">
         <event id="spine_03_ev_01" />
         <event id="spine_03_ev_02" />
         <event id="spine_03_ev_03" />
         <event id="spine_03_ev_04" />
      </spine>
      <event_mappings>
         <event_mapping>
            <event spine_id_ref="spine_01" event_id_ref="spine_01_ev_01" />
            <event spine_id_ref="spine_02" event_id_ref="spine_02_ev_01" />
            <event spine_id_ref="spine_03" event_id_ref="spine_03_ev_01" />
         </event_mapping>
         <event_mapping>
            <event spine_id_ref="spine_01" event_id_ref="spine_01_ev_02" />
            <event spine_id_ref="spine_02" event_id_ref="spine_02_ev_01" />
            <event spine_id_ref="spine_03" event_id_ref="spine_03_ev_04" />
         </event_mapping>
      </event_mappings>
   </spines>
   <layer id="layer_01" name="notational" description="Here goes the description">
      <layer_instance id="layer_01_first_score" order="1" description="first score">
         <layer_instance_part description="layer_01_first_score_01" order="1" src="score1_01.jpg">
            <event_description spine_id_ref="spine_01" event_id_ref="spine_01_ev_01">
               <position><!-- Layer-dependent definition of position --></position>
            </event_description>
            <event_description spine_id_ref="spine_01" event_id_ref="spine_01_ev_02">
               <position>...</position>
            </event_description>
            ...
         </layer_instance_part>
         <layer_instance_part description="layer_01_first_score_02" order="2" src="score1_02.jpg">
            <event_description spine_id_ref="spine_01" event_id_ref="spine_01_ev_06">
               <position>...</position>
            </event_description>
            ...
         </layer_instance_part>
         <layer_instance_part description="layer_01_first_score_03" order="3" src="score1_03.jpg">
            <event_description spine_id_ref="spine_01" event_id_ref="spine_01_ev_07">
               <position>...</position>
            </event_description>
         </layer_instance_part>
      </layer_instance>
      ...
   </layer>
   <layer id="layer_02" name="audio" description="Here goes another description">
      ...
   </layer>
   ...
</ieee1599>
```

Figure 3. Minimal working example of the changes introduced with v2.0.

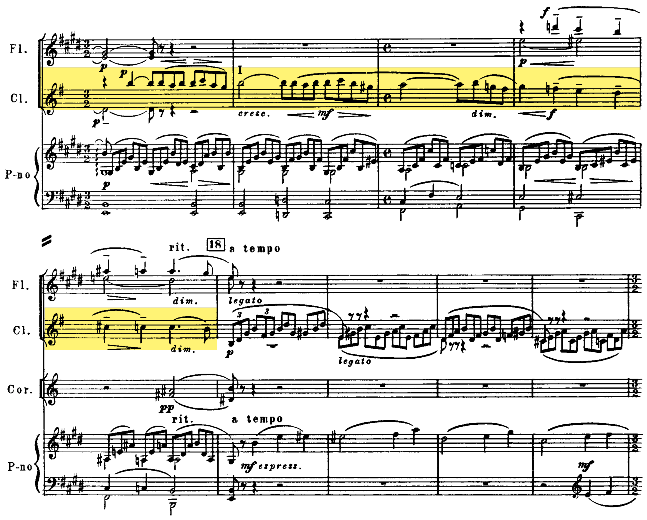

Figure 4. Excerpt from Piano Concerto No.2, Op.18 – 2nd movement "Adagio sostenuto" by Sergei Rachmaninoff. The part of interest has been highlighted.

Figure 5. Excerpt from "All By Myself" by Eric Carmen.

6. Conclusions and Future Works

A question worth asking is "How is the format changing?" The answer is that IEEE 1599 is striving to evolve toward a more symmetric, customizable, and inclusive format.

Moreover, IEEE 1599 is unveiling potential as a general-purpose synchronization format for time-based and space-based information. In this sense, music is a good test bed, as a comprehensive description of a piece may cover different domains and heterogeneous media types. From another perspective, the authors are aware of the risk of losing the main objective of the working group and, ultimately, of the format itself, i.e. the focus on musical information.

Although aware that the present discussion does not conclude the work on the format and that in the future yet other new versions might become available, the authors remain confident that the direction in which the format is moving will allow for a simpler implementation of changes, together with wider support for different applications that were not possible with the original version.

The drafts and companion materials are available at https://ieee1599.lim.di.unimi.it/.

References

[1] A. Baratè, G. Haus, and L. A. Ludovico, "State of the Art and Perspectives in Multi-Layer Formats for Music Representation," in *Proceedings of the 2019 International Workshop on Multilayer Music Representation and Processing (MMRP 2019)*. IEEE CPS, 2019, pp. 27–34.

[2] K. Chen, W. Zhang, S. Dubnov, G. Xia, and W. Li, "The effect of explicit structure encoding of deep neural networks for symbolic music generation," in *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*. IEEE, 2019, pp. 77–84.

[3] P. Hitzler, "A review of the semantic web field," *Communications of the ACM*, vol. 64, no. 2, pp. 76–83, 2021.

[4] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of musical things: Vision and challenges," *Ieee access*, vol. 6, pp. 61 994–62 017, 2018.

[5] I. Jang, P. Kudumakis, M. Sandler, and K. Kang, "The mpeg interactive music application format standard [standards in a nutshell]," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 150–154, 2010.

[6] D. L. Baggi and G. M. Haus, *Music navigation with symbols and layers: Toward content browsing with IEEE 1599 XML encoding*. John Wiley & Sons, 2013.

[7] P. Bellini and P. Nesi, "WEDELMUSIC format: An XML music notation format for emerging applications," in *Proceedings First International Conference on WEB Delivering of Music. WEDELMUSIC 2001*. IEEE, 2001, pp. 79–86.

[8] M. Good *et al.*, "MusicXML: An internet-friendly format for sheet music," in *Xml conference and expo*. Citeseer, 2001, pp. 03–04.

[9] M. Good and G. Actor, "Using MusicXML for file interchange," in *Proceedings Third International Conference on WEB Delivering of Music*. IEEE, 2003, p. 153.

[10] A. Baratè, G. Haus, L. A. Ludovico, and D. A. Mauro, "IEEE 1599 for Live Musical and Theatrical Performances," *Journal of Multimedia*, vol. 7, no. 2, pp. 170–178, 2012.

[11] L. A. Ludovico and C. Zambelli, "Web-Based Frameworks for CLIL in Primary School: Design, Implementation, Pilot Experimentation and Results," in *Computers Supported Education - 8th International Conference, CSEDU 2016, Rome, Italy, April 21-23, 2016, Revised Selected Papers*, ser. Communications in Computer and Information Science, G. Costagliola, B. M. McLaren, J. Uhomoibhi, and S. Zvacek, Eds. Springer, 2017, vol. 739, pp. 139–158.

[12] L. A. Ludovico and D. A. Mauro, "Sound and The City: Multi-Layer Representation and Navigation of Audio Scenarios," in *Proceedings of the SMC 2009 - 6th Sound and Music Computing Conference, 23-25 July 2009, Porto - Portugal*, A. Barbosa, F. Gouyon, and X. Serra, Eds. Porto: SMC, 2009, pp. 19–24.

[13] A. Baratè, G. Haus, and L. A. Ludovico, "A Critical Review of the IEEE 1599 Standard," *Computer Standards & Interfaces*, vol. 46, pp. 46–51, 2016.

[14] G. Castan, M. Good, and P. Roland, "Extensible markup language (XML) for music applications: An introduction," *The Virtual Score*, vol. 12, pp. 95–102, 2001.

[15] A. Lindsay and W. Kriechbaum, "There's more than one way to hear it: multiple representations of music in MPEG-7," *Journal of New Music Research*, vol. 28, no. 4, pp. 364–372, 1999.

[16] J. Steyn, "Framework for a music markup language," in *Proceeding of the First International IEEE Conference on Musical Application using XML (MAX2002)*. IEEE Computer Society Los Alamitos, 2002, pp. 22–29.

[17] G. Haus and M. Longari, "A multi-layered, time-based music description approach based on XML," *Computer Music Journal*, vol. 29, no. 1, pp. 70–85, 2005.

```
%G $xF @3/2 4-''4B+{8BBAB}{B'''C''AG} / 2B+{8BBAB}{B'''C''BxG} / @c 2A+{8ABnGF} / 4GnFED / xCnC4.C8'B /
```

Figure 6. Measures 13–17 of the clarinet part encoded with *PEC*.

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-model href="https://music-encoding.org/schema/dev/mei-all.rng" type="application/xml" schematypens="http://relaxng.org/ns/structure/1.0"?>
<?xml-model href="https://music-encoding.org/schema/dev/mei-all.rng" type="application/xml" schematypens="http://purl.oclc.org/dsdl/schematron"?>
<mei xmlns="http://www.music-encoding.org/ns/mei" meiversion="5.0.0-dev">
   <meiHead>...</meiHead>
   <music>
      <body>
         <mdiv xml:id="m4s4uc4">
            <score xml:id="so3qcsy">
               <scoreDef xml:id="sgtp1zt" meter.count="3" meter.unit="2">
                  <keySig xml:id="k6oscn4" sig="1s" />
                  <staffGrp xml:id="s1c77hfv">
                     <staffDef xml:id="s1p91bt0" n="1" lines="5" clef.shape="G" clef.line="2" />
                  </staffGrp>
               </scoreDef>
               <section xml:id="s1wb3e2s">
                  <scoreDef xml:id="svzjjlx" meter.count="3" meter.unit="2">
                     <keySig xml:id="khbsl03" sig="1s" />
                  </scoreDef>
                  <measure xml:id="mfnjo7e" right="single">
                     <staff xml:id="s19obtds" n="1">
                        <layer xml:id="lm40ctx" n="1">
                           <clef xml:id="cmyzslu" shape="G" line="2" />
                           <rest xml:id="r5odh2d" dur="4" />
                           <note xml:id="n16gprxv" dur="4" oct="5" pname="b" />
                           <beam xml:id="b1oxd4nf">
                              <note xml:id="n1w2vfrn" dur="8" oct="5" pname="b" />
                              <note xml:id="nk6xyvl" dur="8" oct="5" pname="b" />
                              <note xml:id="nhghzyb" dur="8" oct="5" pname="a" />
                              <note xml:id="ngx8bjh" dur="8" oct="5" pname="b" />
                           </beam>
                           <beam xml:id="b23h8om">
                              <note xml:id="n82fpt" dur="8" oct="5" pname="b" />
                              <note xml:id="n3elz0m" dur="8" oct="6" pname="c" />
                              <note xml:id="n1bb0tuc" dur="8" oct="5" pname="a" />
                              <note xml:id="n6sh6uc" dur="8" oct="5" pname="g" />
                           </beam>
                        </layer>
                     </staff>
                     <tie xml:id="t1yevvwh" startid="#n16gprxv" endid="#n1w2vfrn" />
                  </measure>
                  <measure xml:id="m1k4iy3e" right="single">...</measure>
                  <scoreDef xml:id="s16zdwlv" meter.sym="common" />
                  <measure xml:id="mzosgrv" right="single">...</measure>
                  <measure xml:id="mshell2" right="single">...</measure>
                  <measure xml:id="m1iiegxs" right="single">...</measure>
               </section>
            </score>
         </mdiv>
      </body>
   </music>
</mei>
```

Figure 7. Measure 13 of the clarinet part encoded with *MEI*. For the sake of brevity, measures 14–17 are only sketched.

[18] A. Baratè, G. Haus, L. A. Ludovico, and P. Perlasca, "Managing Intellectual Property in a Music Fruition Environment - The IEEE 1599 Case Study," *IEEE MultiMedia*, vol. 23, no. 2, pp. 84–94, 2016.

[19] P. Roland, "XML4MIR: Extensible Markup Language for Music Information Retrieval." in *ISMIR*, 2000.

[20] A. Hankinson, P. Roland, and I. Fujinaga, "The Music Encoding Initiative as a Document-Encoding Framework." in *ISMIR*, 2011, pp. 293–298.

[21] R. Viglianti, "A Specification for Addressing Encoded Music on the Web," in *Music Encoding Conference Proceedings 2015, 2016 and 2017*, G. Di Bacco, J. Kepper, and P. Roland, Eds. Bavarian State Library (BSB), 2019, pp. 47–49.

[22] E. Parada-Cabaleiro and Á. Torrente, "Preventing Conversion Failure across Encoding Formats: A Transcription Protocol and Representation Scheme Considerations [Poster]," in *Music Encoding Conference Proceedings 2020*, E. De Luca and J. Flanders, Eds. Humanities Commons, 2020, pp. 105–108.

[23] W. Goebl and D. M. Weigl, "Alleviating the Last Mile of Encoding: The *mei-friend* Package for the Atom Text Editor," in *Music Encoding Conference Proceedings 2021*, S. Münnich and D. Rizo, Eds. Humanities Commons, 2022, pp. 31–39.

[24] A. Berndt, "Music Performance Markup: Format and Software Tools Report," in *Music Encoding Conference Proceedings 2021*, S. Münnich and D. Rizo, Eds. Humanities Commons, 2022, pp. 57–63.