12-9-2010

Automatic media segmentation within IEEE 1599–2008

Antonello D'Aguanno

Luca A. Ludovico

Davide Andrea Mauro
Marshall University, maurod@marshall.edu

Automatic Media Segmentation within IEEE 1599-2008

Antonello D'Aguanno, Luca A. Ludovico, and Davide A. Mauro
Laboratorio di Informatica Musicale (LIM),
Dipartimento di Informatica e Comunicazione (DICo)
Università degli Studi di Milano
Via Comelico, 39/41 - I-20135 Milano, Italy
{daguanno, ludovico, mauro}@dico.unimi.it
http://www.lim.dico.unimi.it/

Abstract

This paper deals with the automatic extraction of synchronization data from IEEE 1599-2008, an XML-based standard aiming at a comprehensive description of music. Within such format, audio tracks and video contents related to the same music piece can be referred to the occurrence of symbolic music events. In this way, digital objects are mutually synchronized, too. The goal is to show how timing information can be easily extracted from an IEEE 1599-2008 file, converted into a suitable format, and finally employed in a multimedia editing environment in order to produce an automatic segmentation of media objects.

1 Introduction

In the field of computer music and multimedia entertainment, it can be useful to have both symbolic and multimedia information within a unique format. In this way, the same piece can be described in terms of music symbols (e.g. chords, rests, etc.) as well as audio and visual contents (e.g. tracks, video clips, etc.). Needless to say, an integrated description would allow to relate the logical representation of each music event to its multimedia rendering. This is an important feature for a number of software applications: for instance, the products aiming at score following, which let the user watch either a score or one of its visual representations (a choreography, an opera, a television take) while he/she is listening at the corresponding audio content. In this sense, the references among media objects and symbolic events provide the synchronization among different media.

However, another class of applications can be created on the base of such mapping. Usually, a single score can present a number of different performances. Let us consider a genre where performances are usually literal: even for pieces of (so-called) classical music, a high number of versions is available. Another relevant example is given by jazz music, where improvisation on a well known harmonic grid and a systematic use of extemporary instrumental solos originate a high number of variations. Finally, even genres like pop and rock music often produce many versions of the same score: the original piece, the radio edit, one or many live exhibitions, singles, unplugged versions, etc. For music where many versions are available and referable to a single score, the synchronization of events within media objects with the symbolic events of the score virtually allows to switch from a version to another in real time. As a consequence, a richer description of the music piece can be provided, where different audios and videos are linked.

Among other applications, let us cite the possibility to listen to the vocal performances of different artists singing the same *aria*, the possibility to enjoy the differences among different orchestrations of the same piece, the possibility to compare different choreographies for the same ballet, and so on.

In this work, another application field will be explored, namely the automatic extraction of information to create markers and areas within a media object on the base of synchronization data. Intuitively, if a music event is both encoded in terms of symbolic description and located in one or many media files, this information can be used to drive the process of segmentation of the corresponding digital objects.

All this is made possible by a music format where not only symbolic aspects are defined, but also multimedia objects are supported, and music events are described as symbols as well as they are recognized within digital files.

2 Key Features of IEEE 1599-2008

IEEE 1599-2008 is a format to represent data and metadata referable to the same music piece. This project proposes to represent music symbolically in a comprehensive way, opening up new ways to make both music and music-related information available to musicologists and performers on one hand, and to non-practitioners on the other. Its ultimate goal is to provide a highly integrated representation of music, where score, audio, video, and graphical contents can be appreciated together [2]. The key characteristics of this format are:

- Richness in multimedia descriptions for the same music piece - symbolic, logic, graphic, audio, and video contents;
- For each type of multimedia description, possibility of linking a number of digital objects - for instance, many performances of the same piece, or many score scans from different editions;
- Full support for synchronization among time-based contents. For instance, audio and video contents can be synchronized as the score advances while music is being played, even when switching from a particular performance to another.

2.1 Multi-layer Structure

As stated before, a comprehensive description of music must support heterogeneous materials. Thanks to the intrinsic capability of XML to provide strongly structures for information, such representations can be organized in an effective and efficient way. IEEE 1599-2008 employs six different layers to represent information:

- *General* - music-related meta data, i.e. catalogue information about the piece;
- *Logic* - the logical description of score symbols;
- *Structural* - identification of music objects and their mutual relationships;
- *Notational* - graphical representations of the score;
- *Performance* - computer-based descriptions and executions of music according to performance languages;
- *Audio* - digital or digitized recordings of the piece.

Not all layers must, or can, be present for a given music piece. Of course, the higher their number, the richer the musical description. Richness has been mentioned in regard to the number of heterogeneous types of media description, namely symbolic, logic, audio, graphic, etc. But the philosophy of IEEE 1599-2008 allows one extra step, as each layer can contain many digital instances. For example, the *Audio* layer could link to several audio tracks and even videos for the same piece. The concept of *multi-layered description* - as many different types of descriptions as possible, all correlated and synchronized - together with the concept of *multi-instance support* - as many different media objects as possible for each layer - provide rich and flexible means for encoding music in all its aspects.

Moreover, it is possible to adopt some *ad hoc* encoding in addition to already existing formats to represent information. In fact, while a comprehensive format to represent music is not available, popular existing standards must be taken into account. This is a not a contradiction because of the two-sided approach of IEEE 1599-2008 to music representation, which is: keep intrinsic music descriptions inside of the IEEE 1599-2008 file - in XML format - and media objects outside of the IEEE 1599-2008 file - in their original format.

Consider the following examples. The symbols that belong to the score, such as chords and notes, are described in XML, in the *Logic* layer. On the contrary, MP3 files and other audio descriptions are not translated into XML format, rather they are linked and mapped inside the corresponding IEEE 1599-2008 layer, the *Audio* layer.

2.2 The Spine

In this approach, heterogeneous descriptions of the same music piece are not simply linked together, but a further level of detail is provided: whenever possible, media information is related to single music events. The concept of music event is left intentionally vague, since the format is flexible and suited to many purposes. A music event can be defined as the occurrence within the score, or its abstraction, of something that is considered relevant by the author of the encoding. A standard case is considering all notes and rests of a score as music events. From a more general standpoint, all symbols in a score could be music events, ranging from clefs to articulation signs. Also, for particular purposes, other interpretations are allowed. For instance: in jazz music the event list could correspond to the harmonic grid; in dodecaphonic music, an event could be the occurrence of a series; in segmentation, music events could be the starting and ending points of music objects.

The *spine* consists of a sorted list of events, where the definition and granularity of events can be chosen by the author of the encoding. The *spine* has a fundamental theoretical importance within the format. It represents an abstraction level, as the events identified there do not have to correspond to score symbols, or audio samples, etc. It is the author who can decide, from time to time, what goes under the definition of music event, according to the needs.

Since the *spine* simply lists events to provide a unique label for them, the mere presence of an event in the *spine* has no semantic meaning. As a consequence, what is listed in the *spine* structure must have a counterpart in some layer, otherwise the event would not be defined and its presence in the list (and in the IEEE 1599-2008 file) would be of no relevance. For example, in a piece made of n music events, the *spine* would list n entries without defining them from any point of view. Now, each *spine* event can be described:

- in 1 to n layers; e.g., in the *Logic*, *Notational*, and *Audio* layers;
- in 1 to n instances within the same layer; e.g., in three different audio clips mapped in the *Audio* layer;
- in 1 to n occurrences within the same instance; e.g., the notes in a song refrain that is performed 4 times (thus the same *spine* events are mapped 4 times in the *Audio* layer, at different timings).

Thus, the events listed in the *spine* structure can correspond to one or to many instances in other layers. This aspect creates synchronization among instances within a layer (*intra-layer synchronization*), and also synchronization among contents disposed in many layers (*inter-layer synchronization*).

3 Synchronization Mechanisms

In order to get synchronization among IEEE 1599-2008 layers, a mechanism based on the *spine* concept has been introduced. *Spine* allows the interconnection of these layers on space and time domain through a relative measure in *spine* and an absolute measure in the other layers.

Inside *spine*, each music event is univocally defined by an identifier (the `id` attribute) and carries information about timing and position. Timing (the `timing` attribute) is expressed in a relative way: the measurement unit is user-defined in function of time domain and its value is the distance from the preceding event. For instance, a quarter note may correspond to 1024 timing units, no matter which absolute timing it has in an audio recording. Please note that the absolute timing of a music event depends on the performance, so it will be described in the *Audio* layer. The `hpos` attribute, standing for horizontal position, has a similar meaning referred to space domain. A simplified example of *spine* could be the following one:

```
<spine>
   <event id="e1" timing="0" hpos="0"/>
   <event id="e2" timing="1024" hpos="5"/>
   <event id="e3" timing="512" hpos="10"/>
   ...
</spine>
```

In the previous example, three music events are listed within the *spine* structure. The second event occurs 1024 time units after the first one, whereas the third occurs 512 time units after the second one. We have to remark two key points: i) such values are theoretical, and ii) those values have no absolute meaning, as neither physical time units nor rhythmical music values are directly involved in their *spine* definition.

The approach is completely different in the *Audio* layer, where every media linked to IEEE 1599-2008 is mapped to *spine* events through the `track` tag. This element is a container for a number of `track_event` elements, which contain the *spine* identifier (the `event_ref` attribute) and absolute references (the `start_time` and `end_time` attributes) that specify the absolute occurrence of the event in the media file. Through this mechanism, each single music event in *spine* can be physically indexed and recognized within one or many digital objects.

Unlike *spine* timing, the *Audio* layer contains absolute time references, allowing the use of different measurement units in function of the various kinds of media. As regards the `timing_type` attribute, the default unit is the second, but it is possible to use bytes, samples, or frames as well. This solution provides a more sophisticated time granularity according to the different media structures (AAC, MP3, PCM, etc.) supported by IEEE 1599-2008.
A simplified example of the *Audio* layer contents could be the following one:

```
<audio>
  <track file_name="audio/example.mp3"
         encoding_format="audio_mpeg"
         file_format="audio_mpeg">
    <track_indexing timing_type="seconds">
      <track_event event_ref="e1"
              start_time="0.00" />
      <track_event event_ref="e2"
              start_time="1.15" />
      <track_event event_ref="e3"
              start_time="1.67" />
      ...
    </track_indexing>
  </track>
</audio>
```

This paper does not address specifically the techniques to get synchronization among layers. In general terms, synchronization in time domain can be obtained following three different methodologies.

- Hand-made approach: synchronization and *spine* are completely created by human work. For this purpose, it is necessary a good experience both as a musician and as a trained listener.
- Semi-automatic approach: main beats are manually

set, whereas intermediate notes are looked for by *ad hoc* algorithms based on interpolation techniques.

- Automatic approach: every music event is recognized automatically by means of audio-score synchronization algorithms.

The last approach is the most interesting, as it does not require human supervision, thus it is possible to process a huge amount of pieces in a short lapse of time. The state of the art about automatic audio-score synchronization algorithms provides different approaches, but most algorithms are performed in two steps: i) audio and score analysis, and ii) identification of links between the two layers [5].

Different algorithms are proposed in literature to implement audio analysis with well-known tools from audio signal processing. For example, in [7] a Short Time Fourier Transform is computed, while in [1] an onset detection followed by pitch detection is employed. In [6] a decomposition of the audio signal into spectral bands related to fundamental and harmonic pitches is effectuated. For each band the positions of significant energy increases is calculated; such positions are candidates for note onsets. However, the most accepted solution in literature is a template-matching technique, that is used to select the correct links between audio and score ([4], [8]). Such algorithms render a MIDI score to obtain a template of the real execution, then the result is compared to the actual audio, often by employing a DTW[1] programming technique.Despite the efforts to obtain automatic synchronization and the various algorithms proposed to address this problem, actually there is not a commonly accepted solution for this problem in a general case. However, for our purpose, it is interesting to describe the representation of synchronization data, and not the way they are obtained.

Now we provide an evolution of the code block previously introduced. It is an IEEE 1599-2008 example of *Audio* layer containing two tracks, where the default time units are the second and the sample-aligned frame, respectively.

```
<audio>
  <track file_name="audio/example.mp3"
         encoding_format="audio_mpeg"
         file_format="audio_mpeg">
    <track_indexing timing_type="seconds">
      <track_event event_ref="e1"
              start_time="0.11" />
      <track_event event_ref="e2"
              start_time="1.15" />
      <track_event event_ref="e3"
              start_time="1.67" />
      ...
    </track_indexing>
  </track>
  <track file_name="video/example.mpg"
         encoding_format="video_mpeg"
         file_format="video_mpeg">
    <track_indexing timing_type="frames">
      <track_event event_ref="e1"
              start_time="10" />
      <track_event event_ref="e2"
              start_time="45" />
      <track_event event_ref="e3"
              start_time="63" />
      ...
    </track_indexing>
  </track>
</audio>
```

The code shows how, through the same structures, it is also possible to synchronize videos containing musical contents (e.g. the soundtrack of a movie or the video clip of a song in MPEG format). Once again, the synchronization is based on absolute timing values.

Please note that each music event is logically mapped on *spine*, and it is accordingly referenced from other layers or other representations inside the same layer. In this approach, identifiers are fundamental to jump from a kind of representation to another, and to allow synchronized movement along layers both in time and in space domain.

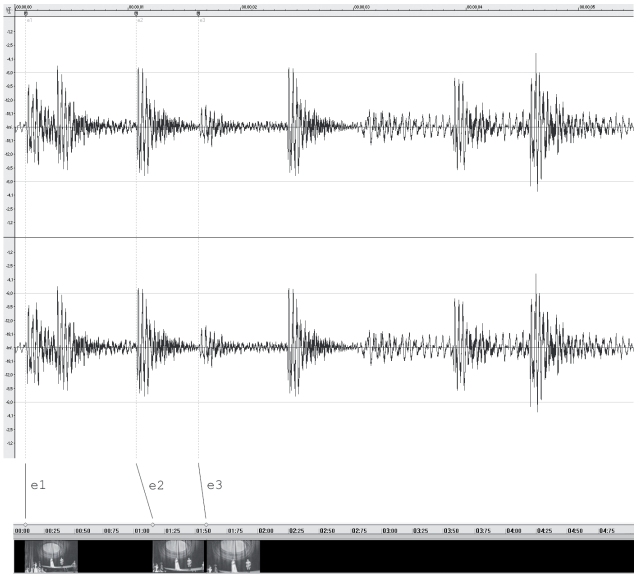

Figure 1. Music events in two different digital objects. The events are graphically shown as markers in a popular audio editing tool and as keyframes in a DVD authoring software.

Figure 1 illustrates the mentioned device applied to an

[1] DTW, standing for Dynamic Time Warping, is a technique for aligning time series that has been in use in the speech recognition community since the 1970's.

audio and a video, both included in a single IEEE 1599-2008 file. As intuitively shown by vertical lines, *spine* mechanism allows not only to investigate all the encoded representations of the same *spine* events, but also to jump logically from a layer to another by finding correspondences inside heterogeneous contents.

4 Extraction of Synchronization Data

In the multimedia field there are many software applications which are enabled to use, store and share lists of time anchors. These lists can be employed in many ways: for example it is possible to split a movie into chapters by using DVD-authoring software or to label an audio file in order to obtain a synchronization between audio contents and the related music symbols.

Currently, there do not exist not commonly accepted file formats to save both symbolic information and audio-related synchronization for future computations. In this paper we focus on the IEEE 1599-2008 capabilities to solve this problem. In IEEE 1599-2008, synchronization anchors can be inserted in three different layers: *Notational*, *Performance* and *Audio*. In particular, inside the *Audio* layer there is the list of timestamps related to the audio/video files (see Section 3). Thanks to XML capabilities, this information can be easily accessed and parsed, which allows to design and implement software tools to export it according to other in-use standards. For instance, an application can read this list and translate its contents in a formatted plain text which can be imported into a multimedia authoring program.

It is worth to cite two practical examples. First, most audio editing tools can import files of formatted text to specify markers in a file. This is the case of Sony Sound Forge, that expects a list of entries in one of the many supported formats: time (`hr.mn.sc,xxx`), time & frames (`hr:mn:sc.fr`), measures & beats (`ms:bt,qbt`), etc. In general, at the end of the import process markers can be saved inside audio formats. The other relevant example involves the automatic creation of chapters in a DVD authoring environment. This case will be described in some detail in the next section.

The software we have released reads the entire *Audio* layer and separates the time-related information from event name and other XML syntax. For example the event

```
<track_event event_ref="e1"
         start_time="1.67"/>
```

can originate a text line such as `1.67` or `00:00:01.670`, depending on the required format. By performing this operation on a whole track, we obtain a sequence of formatted time data. Furthermore, this list can be translated again in order to accomplish the time format of other programs.

Please note that time information could have different granularity; for example in an MP3 the granularity is related to frame dimension and in a CD-DA to samples. In a DVD - using MPEG2 compression standard - is possible to set chapter starting-time in a less accurate way because it has to be attached to an I-frame[2], and typically an I-frame occurs every 15 frames (NTSC transmits 30 frames/s).

5 Case Study

The approach described in the previous sections has inspired the creation of a multimedia entertainment product, realized at the *Laboratorio di Informatica Musicale (LIM) - Università degli Studi di Milano*. The symbolic contents of the pop song *Non credere*, by Mina (Anna Maria Mazzini, a famous italian singer), were encoded in IEEE 1599-2008 format together with four different performances and lyrics. All these media have been semi-automatically synchronized.

The final goal was obtaining a DVD-video where the user could skip to different performances of the same piece, or jump to a particular verse of the lyrics in a synchronized environment. As regards audio, the DVD-video format allows various tracks for multi-language support; but, the standard was not conceived to support video synchronization among a number of parallel clips. Thus, a number of synchronization points had to be set in order to jump from a video clip to another. In particular, the result was obtained by splitting the original video materials into a number of chapters. As synchronization data are already present inside the XML file, they were used to drive the process. The operations to be performed can be summarized as follows:

- Reading all the synchronization events related to a particular performance;
- Translating this information into a plain-text list;
- Importing this list into a DVD-authoring system;
- For each video, automatically splitting the movie into chapters.

The authoring system used in this project was *Mediachance DVD-Lab Pro*. Figure 2 illustrates the design phase.

As stated in the previous section, the granularity of DVD chapters cannot be very accurate in time dimension, due to the I-frame mechanism. Besides, the controls to navigate DVD contents are not comparable to those provided by a computer interface [3]. Thus, a selection of relevant events

[2] An Intra-frame (I-frame) is a compressed version of a single uncompressed frame. It does not depend on data in the preceding or the following frames and it is the only frame kind which a reproduction can start.

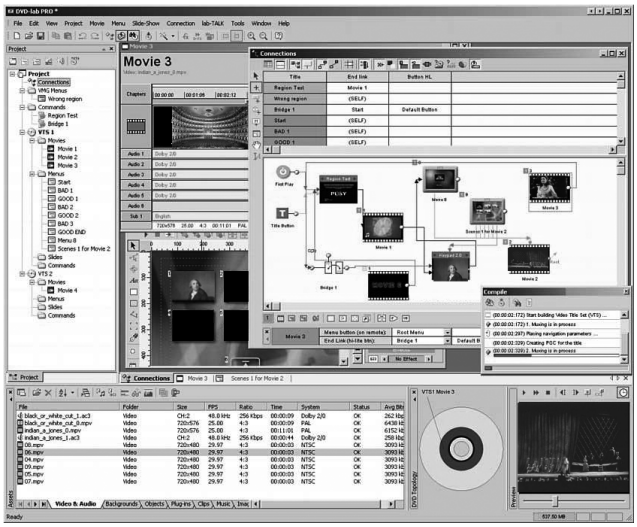

Figure 2. The design phase of the DVD product using *Mediachance DVD-Lab Pro***.**

had to be individuated. From the *Audio* layer the software extracted only the timestamps related to the first word of each verse. Time-related fixed-point values were translated into the following format: hh:mm:ss:ccc, as required by *DVD-Lab Pro*. By repeating this process for each audio/video track, a complete synchronization was obtained.

From the user's point of view, this DVD allows to jump from chapter to chapter (namely from a lyrics line to another) in a conventional way, like scenes in a standard movie; but a number of additional on-screen buttons provide the controls to switch the audio/video contents currently playing nearly in real time. Videos are originally aligned on the first note, namely the song's *incipit*, but their timing can differ because of a slower or faster performance. However, 3 of them are playback executions, thus music

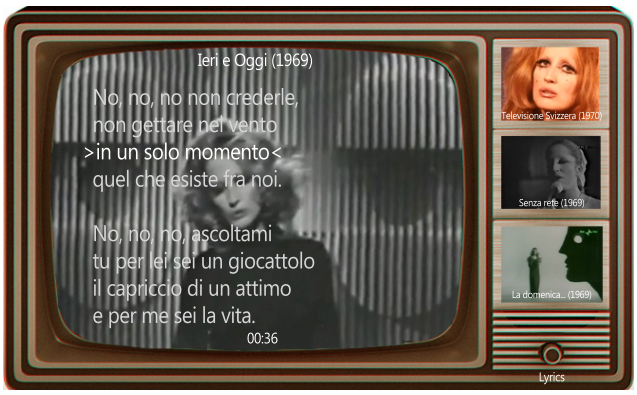

Figure 3. Screenshot of the DVD-Video.

events occur at the same time. As regards the fourth video, it is time-stretched with respect to the length of other videos. Due to DVD-video constraints, all the materials related to the song must be included into a single video track. As a consequence, the final product presents 4 titles (magnifying each clip), both with and without overprinted lyrics. Please note that lyrics cannot be treated through the subtitle feature, since they represent a navigation tool and should be selectable by the user (see Figure 3).

6 Conclusions and Future Works

In this paper we analysed the reliability of IEEE 1599-2008 in representing synchronization data and its capability to act as a specialized container for such information. Timing information can be easily and automatically extracted, converted in any suitable format, and finally used to produce segmentation in a media object. As regards future works, the algorithms used in our application should be refined in order to obtain a reliable automatic synchronization among audio/video contents and *spine* events.

This is only one of the possible applications of the IEEE 1599-2008 format, but we consider it particularly interesting for the market of multimedia home entertainment.

References

[1] V. Arifi, M. Clausen, F. Kurth, and M. Muller. Automatic synchronization of music data in score-, midi-and pcm-format. In *4th International Conference on Music Information Retrieval, ISMIR 2003*, 2003.

[2] D. Baggi and G. Haus, editors. *Proceedings of the IEEE CS Conference The Use of Symbols To Represent Music And Multimedia Objects.*, Lugano, 2008. IEEE CS.

[3] A. Baratè and L. Ludovico. Advanced interfaces for music enjoyment. In *Proceedings of the International Working Conference Advanced Visual Interfaces (AVI 2008)*, Napoli, Italy, 2008. ACM Press.

[4] R. B. Dannenberg and N. Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proceedings of the 2003 International Computer Music Conference*, 2003.

[5] S. Dixon and G. Widmer. Match: A music alignement tool chest. In *6th International Conference on Music Information Retrieval, ISMIR 2005*, 2005.

[6] M. Muller, F. Kurth, and T. Roder. Towards an efficient algorithm for automatic score-to-audio synchronization. In *5th International Conference on Music Information Retrieval, ISMIR 2004*, 2004.

[7] F. Soulez, X. Rodet, and D. Schwarz. Improving polyphonic and poly-instrumental music to score alignment. *4th International Conference on Music Information Retrieval, ISMIR 2003*, pages 143–148, 2003.

[8] R. Turetsky and D. Ellis. Ground-truth transcriptions of real music from force-aligned midi syntheses. *4th International Conference on Music Information Retrieval, ISMIR 2003*, 2003.