

MUSIC AND GESTURE FILE: PERFORMANCE VISUALISATION, ANALYSIS, STORAGE AND EXCHANGE

Stuart Pullinger; Douglas McGilvray; Nicholas J. Bailey
Centre for Music Technology
University of Glasgow

ABSTRACT

To enable users from outside the music information retrieval and programming community to access the large amounts of data which has been collected in databases, and to aid in interdisciplinary collaboration, experience at the University of Glasgow has shown that data must be presented in an informative yet accessible fashion. New rich interactive interfaces to musical performance data require multiple views of related data: score, audio, video etc. The nature of interdisciplinary research means that collaborators are often situated in disparate locations using many different tools and systems. The current work aims to integrate data in formats which are already in use and which applications already support. Performance Markup Language (PML) offers the capability to integrate this data. It operates at the level above data storage, at the level of data integration, enabling gestures and performance events expressed in many disparate data sources to be consolidated into a single entity. By combining PML and the data sources into a widely used compression format, we encapsulate the multifarious data into an easily transported form whilst preserving the accessibility of that data.

1. INTRODUCTION

Much research time has been devoted to the topic of representation of musical data [5] [6].

Many representations and formats exist for music and gesture data: gestural data can be stored in GMS [4] or GDIF [7]; scores can be represented in MusicXML [3] or MEI [8]; audio can be stored in wave or aiff or compressed in mp3 or Ogg/Vorbis. It is not the objective of the current work to add to this list but to seek a way to integrate such data thus enabling the development of a new class of rich interactive tools for the analysis of music and performance.

Previous efforts in the field such as SMDL [9] have attempted to prescribe new formats which have suffered from a lack of tools to support them [10]. The current work aims to avoid this by using existing formats - where tools are already available, and only specifying where necessary for integration.

This paper introduces the Music and Gesture File (MGF) - a container format for the storage and exchange of musical scores and matched performance data including audio,

video and gestural data. The format is under development at the Centre for Music Technology, University of Glasgow. More information, examples and tools can be found here [1].

The Centre has participated in projects, such as [14], which required the storage and exchange of musical scores alongside matched performance data. To achieve this, the Performance Markup Language (PML) (see Section 2.1.2) was developed to combine a MusicXML score with score-matched performance data in a single XML file. Ongoing projects such as [2] require the storage and exchange of multi-modal performance data - combining a score with audio, video and gestural data from motion capture which cannot easily be contained in an XML file. The current work aims to facilitate this and future work by defining a new file format which allows score and score-matched multi-modal performance data to be stored and exchanged efficiently and easily.

1.1. Requirements

The requirements for the container format for musical performance fall into these broad categories:

1. Use existing formats: Where possible the file should contain data in already defined formats to avoid duplicating the effort of others by defining a new file format.
2. Use standard and open formats: The data should be carried in formats which are standardised and freely available. This will enable the data to be easily integrated with existing software for minimal cost.
3. Use unencumbered formats: The data should not be held in formats which require patent licenses to access or develop software as this would increase the cost of development.
4. Compressed: The file should use lossless compression (or where appropriate, lossy compression) to save bandwidth on transportation.
5. Standard tools: Where possible, the data should be available to standard tools. For example, XML will be preferred over alternative representations.

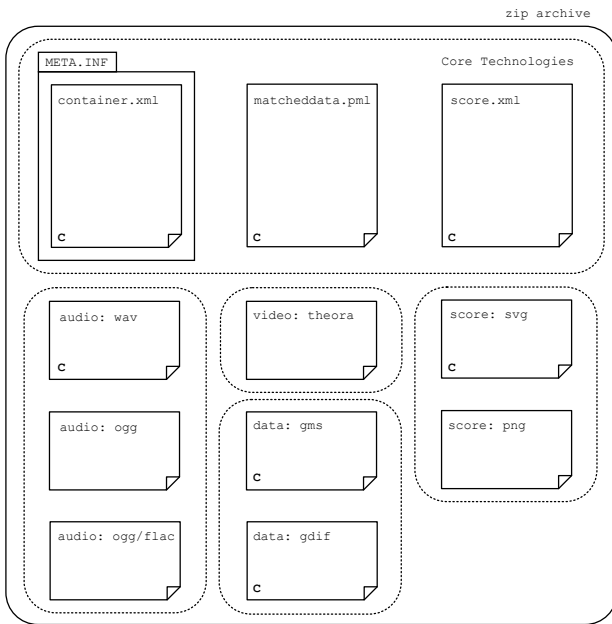


Figure 1. The structure of an MGF file showing the core technologies and preferred representations. Files marked with a C should be compressed when added to the archive.

2. DEFINITION OF A MUSIC AND GESTURE FILE

The proposed container format extends the MusicXML 2.0 specification [11] to include data from other sources inside a compressed archive. (It is worth noting that while additional data is added to the file, it remains a valid MusicXML 2.0 file). The data is integrated through the inclusion of a Performance Markup Language (PML) file (see section 2.1.2). There follows an examination of these core technologies and suitable representations in other domains and the method of their integration with Music and Gesture Files (MGF).

2.1. Core Technologies

A basic MGF file consists of a MusicXML file, a Performance Markup Language (PML) file which references it and a META.INF folder containing a “container.xml” file. These files are compressed into a ZIP archive [23]. For music with no score the MusicXML file can be empty. An overview of the structure of an MGF file is given in Figure 1.

2.1.1. MusicXML 2.0

The MusicXML 2.0 specification [11] extends the more widely used MusicXML 1.1 specification introduced here [3]. Of particular interest to the current work is the compressed format which stores data in a JAR archive [12] (compatible with the popular zip format [23]) with the addition of an index file called container.xml under the META.INF/ folder. The JAR container allows for the inclusion of files which are not MusicXML files. (Where a

file is already compressed, this file should be added to the archive as-is: that is to say, it should not be compressed a second time using zip compression. Where a file is stored in an uncompressed format, for example XML data, the file should be compressed in the zip archive.)

The use of zip/JAR as the container format is particularly attractive since it is widely supported in file browsers and operating systems. This enables access to the contained data even where there is no software which supports MusicXML 2.0 or MGF. It also simplifies the extension of software to support the file format.

2.1.2. Performance Markup Language

The Performance Markup Language (PML) [13] [6] is an XML language developed at the Centre for Music Technology at Glasgow University for the exchange score-aligned performance data between applications. It has been successfully deployed in several projects including [14] and [22]. The annotated PML Document Type Definition (DTD) can be downloaded from here [15].

PML enables the creation of multiple, overlapping hierarchies of gestural information using relational links to XML entities and locations in external media.

A PML document consists of 2 sections: score and performance. The score section can incorporate a score in an XML format such as MusicXML or MEI or it can reference the contents of an external file. For the current application it would reference an external MusicXML file.

The performance section contains the data pertinent to one performance and can include one or more performance parts, each containing data regarding the performance of one performer. Each performance section and performance part contains metadata relating to the time, location, performer etc. The performance part contains *sound events* and *gesture events*. All events have a start time and optionally a duration. Time can be expressed in seconds or in the time base of an external medium. For example, a single gestural event can be defined as occurring at an exact frame in a video stream and at an exact sample frame in an audio file with a completely different frame rate.

By providing a single location for the representation of performance events, PML integrates data from multiple sources allowing for the creation of new interactive presentations of performance data.

2.2. The Visual Domain

2.2.1. Vector Graphics

The preferred visual representation of a score will be in Scalable Vector Graphics (SVG) format [16]. SVG is an XML format describing vector graphics standardised by the W3C. It is supported by most web browsers either natively (Firefox, Opera, Safari, Konqueror) or via a plug-in (Internet Explorer). It can be generated by the Lilypond music typesetting program [17] and is convertible to and from postscript and pdf formats. The standard supports

the inclusion of namespaced data inside a `<metadata>` tag thus allowing the inclusion of PML data. In this manner, performance and gestural and data can be included inside the score from which the performance derives.

2.2.2. Raster Graphics

The preferred visual representation of raster graphics will be Portable Network Graphics (PNG). PNG is a losslessly compressed image format that is exported by most graphics applications and supported by all modern web browsers. The standard (Section 11.3.4.5 [18]) enables the inclusion of internationalised textual data in a 'iTxt' chunk. This is the preferred location for PML data which should be included in compressed form.

2.3. The Gestural Domain

The format for storage of gestural data is a matter of some debate [6]. There is no single format which can be singled out yet as offering wider support in applications. Currently, the preferred representation of gesture is the Gesture and Motion Signal (GMS) file [4] though support for GDIF [7] is planned. GMS stores motion data in mono-dimensional tracks. The tracks are grouped into channels to represent several dimensions. The channels are grouped into units representing a group of channels which are dynamically related - such as all the channels pertaining to one performer. Finally all the units are grouped into a scene which determines the sample rate.

PML supports the integration of data in GMS files through the `<gmsres>` (GMS Resource) and `<gmsref>` (GMS Reference) tags which use the scene/unit/channel/track/frame structure of the GMS file to locate data.

2.4. Audio

2.4.1. Uncompressed Audio

PML supports the integration of audio in Wave format through the `<wavres>` (WAV resource) and `<wavref>` (WAV reference) tags. The `<wavref>` element locates data in a wav file through the channel and framenumbers.

2.4.2. Lossy Audio Compression

The preferred representation for compressed audio will be Ogg/Vorbis. Whilst not the most widely supported audio compression format (that prize would go to mp3) Ogg/Vorbis offers comparable audio quality and file size to mp3. The codec has freely available software for encoding and decoding which recommends it over mp3.

The Vorbis codec compresses audio data which is then encapsulated into an Ogg multiplexed data stream. The Ogg stream allows for the inclusion of several *logical* streams inside a single *physical* Ogg stream. The streams can be continuous (as with audio) or discontinuous (as with PML data). To include PML data, each performance part is encoded into a different stream with complete (ie. opened and closed) event tags occurring at the granule position

(ogg time unit) which matches the start time of the event. By including opening and closing tags the resulting PML file is made more robust to missing events since a missing part of the stream will not corrupt the entire PML file.

2.4.3. Lossless Audio Compression

The DEFLATE compression algorithm [21] used in JAR files is not optimised for audio data. In order to improve the compression ratio and thereby reduce the filesize a lossless audio codec can be used. The preferred format for lossless compression is Ogg/FLAC. This encapsulates data encoded with the Free Lossless Audio Codec (FLAC) [19] into an Ogg stream thus enabling the inclusion of PML data as described in 2.4.2. The FLAC codec can achieve compression ratios of less than 55% and is freely available on most systems.

2.5. Video

The preferred video representation will be Ogg/Theora. This format encapsulates Theora [20] encoded video into an Ogg data stream thus allowing the integration of audio (see Sections 2.4.2 and 2.4.3), video and PML data into a single stream.

Currently the Theora codec does not support uncompressed video data. When such a codec becomes available, the current application would prefer a freely available codec which encapsulates uncompressed video in an Ogg data stream.

3. FUTURE WORK

It would be interesting to develop MGF into the areas of data presentation. The Open Document Format [24] provides a similar though distinct structure to a Music and Gesture File. Future work will use this similarity to investigate the creation of interactive presentations of MGF data sources allowing the file to serve 3 purposes: as a MusicXML file, as an MGF file and as an ODF file.

Future work in PML will include an analysis section to enable exchange of musical analyses.

4. CONCLUSION

The Music and Gesture File format enables the exchange of multiple musical data sources in a convenient container. By using existing standard formats it allows access to the data to the widest possible community of users. By including Performance Markup Language MGF transcends mere data storage and provides the means to integrate data from multiple sources.

5. REFERENCES

- [1] "The Motion and Gesture File Wiki"
<http://markov.music.gla.ac.uk/cmt-wiki/MGF>

- [2] “Multimodal Analysis of Performance Parameters in Chopin’s B Flat Minor Piano Sonata Op.35” *Presented at DMRN+1 Workshop*, Queen Mary University, London, December 2006.
- [3] Michael Good “Musicxml: An internet-friendly format for sheet music”, *XML 2001 Conference Proceedings*, Orlando, FL, December 2001.
- [4] Annie Luciani, Matthieu Evrard, Nicolas Castagné, Damien Couroussé, Jean-Loup Florens, Claude Cadoz “A Basic Gesture and Motion Format for Virtual Reality Multisensory Applications” *Proceedings of the 1st international Conference on Computer Graphics Theory and Applications*, ISBN: 972-8865-39-2, Setubal (Portugal), March 2006.
- [5] Walter B. Hewlett and Eleanor Selfridge-Field, editors *The Virtual Score* MIT Press, 2001
- [6] Alexander Refsum Jensenius, Antonio Camurri, Nichols Castagné, Esteban Maestre, Joseph Malloch, Douglas McGilvray, Diemo Schwarz, Matthew Wright “The Need Of Formats For Streaming And Storing Music-Related Movement And Gesture Data” *Proceedings of the International Computer Music Conference 2007*
- [7] Tellef Kvifte, Alexander R. Jensenius, Rolf Inge Gody. “Towards a gesture description interchange format” *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME) 2006*.
- [8] Perry Roland “Xml4mir: Extensible markup language for music information retrieval” *ISMIR Proceedings 2000*
- [9] Steven R. Newcomb “Standard Music Description Language complies with hypermedia standard” *Computer*, July 1991
- [10] Donald Sloan “Learning our lessons from smdl” *MAX 2002 : musical application using XML 2002*
- [11] “MusicXML 2.0 Specification” <http://www.recordare.com/xml.html>
- [12] “JAR file format specification” <http://java.sun.com/javase/6/docs/technotes/guides/jar/jar.html>
- [13] “On The Analysis of Musical Performance by Computer” PhD Thesis, University of Glasgow, 2007
- [14] Graham Hair, Ingrid Pearson, Amanda Morrison, Nicholas Bailey, Douglas McGilvray, Richard Parncutt “The Rosegarden Codicil: Rehearsing Music in Nineteen-Tone Equal Temperament” *Scottish Music Review* Volume 1, No. 1, 2007
- [15] “Document Type Definition for the Performance Markup Language” <http://www.n-ism.org/DTD/pml.dtd>
- [16] “Scalable Vector Graphics Specification” <http://www.w3.org/TR/SVG11/>
- [17] Han-Wen Nienhuys and Jan Nieuwenhuizen “Lilypond ... music notation for everyone” <http://lilypond.org/web/>
- [18] “Portable Network Graphics” <http://www.w3.org/TR/PNG/> ISO/IEC 15948:2003 (E)
- [19] “The FLAC Formal Specification” <http://flac.sourceforge.net/format.html>
- [20] “Theora Specification” <http://theora.org/doc/Theora.pdf>
- [21] “The DEFLATE Compressed Data Format (RFC 1951) Specification” <http://www.ietf.org/rfc/rfc1951.txt>
- [22] J. MacRitchie, N. J. Bailey, G. Hair “Multimodal acquisition of performance parameters for analysis of Chopin’s b flat minor piano sonata finale op. 35” *DMRN+1: Digital Music Research Network One-day Workshop 2006* Queen Mary, University of London, 20 December, 2006
- [23] “The Info-Zip file format” <ftp://ftp.uu.net/pub/archiving/zip/doc/appnote-970311-iz.zip>
- [24] “OpenDocument v1.0 Specification” <http://www.oasis-open.org/committees/download.php/12572/OpenDocument-v1.0-os.pdf> ISO/IEC 26300:2006