

Towards a Symbolic/Time-Based Music language based on XML

Goffredo Haus, Maurizio Longari
LIM-DSI
Università degli Studi di Milano
via Comelico, 39
20135 Milano, Italia
+39 02 5835 6222, +39 02 5835 6296
haus@dsi.unimi.it, longari@dsi.unimi.it

ABSTRACT

The representation of Symbolic Music Information is the fundamental element of Music Information Processing. Among the several approaches developed in history the markapped one seems to be the most promising. Starting from SMDL, some XML proposals are briefly analyzed. We present our approach of layered representation of Symbolic Music Information based on the space-time organization of events. Our model splits music information in layers of representation and takes the symbolic one as the gravity center. Moreover, the symbolic layer contains a space-time structure by means of which all other layers are brought together. Then we discuss other XML standard that can be usefully related to the XML representation of music information, such as security and vector graphics standards. Lastly, we deal with some open problems in the field of music processing that might be of interest for our view of symbolic representation.

Keywords

Symbolic Music Representation, XML, SMDL

1 INTRODUCTION

Music Information Processing is a discipline as old as computer science. The representation of Symbolic Music Information is its fundamental element. Early attempts were made with DARMS, MUSIC V and other ASCII-based codes.

The advent of structured text (Standard Generalized Markup Language SGML) has raised the idea of representing this information in human-readable fashion. Structuring such information gives the possibility of

representing several aspects of music and more sophisticated concepts hidden in Music Information. SMDL [25][28] (Standard Music Description Language) was the first attempt in this direction. The only drawback was the lack of ad-hoc tools for its implementation.

Another approach has come from the area of electronic music, namely, Musical Instrument Digital Interface (MIDI) and MIDI files. This file format contains sequences of timed symbolic music events. This is the most popular format used in electronic performances and for exchange of electronic scores. Unfortunately, its symbolic music expressiveness is very poor and it is not sufficient for professional, even semi-professional, music composition. It is, however, interesting that it can be located between performance and symbolic representation.

In the past few years, the synergy between Internet and eXtensible Markup Language (XML) [4] has brought to the fore the representation of content meta-data by means of markapped languages. Symbolic information can be thought as the intrinsic meta-data content of music. The purpose of this work is to explore the possibility of presenting our solution for symbolic music information using XML syntax.

The main concepts of this approach are *layer-subdivision* (spanning from signal to structural) and *space-time constructs*. We give herewith some definition of terms widely used through the article.

SMI: Symbolic Music Information (general, structural and music content information)

Source Material: graphic score images, notation files, digitized registrations, videos

Music Work: the set of source material and its related symbolic representation.

A case study in which these concepts are fully used is the Scala Project [14][16], developed at LIM [39]. In this project, the notion of Spine is the key element for the

integration of scores (space dimension) with audio files (time dimension)[15].

In the next paragraphs, we briefly survey SMDL and several XML applications for the representation of music information. Then we describe the concepts of Layered SMI and its integration with markupp languages. In the last paragraphs, some related topics and open problems are discussed.

2 MARKUPPED SMI

The first approach to the representation of SMI with markup languages was that of SMDL. It was developed as the application of SGML for the representation of music information. Because of the complexity of SGML syntax, and of the fact that it was not diely applied, SGML was abandoned. The concepts of timed/based structures were maintained for the development of the much more successful standard HyTime (Hypermedia Time/Based Description Language [9][23]). In fact, a second review of SMDL was made that included the concepts developed in HyTime framework.

In the past few years, several proposal have been made to represent Symbolic Music Information by means of the XML standard. We will briefly discuss the topics of SMDL and then those of XML representations.

SMDL/HyTime

The Standard Music Description Language (SMDL [24]) was created in 1984 from an idea by Charles Goldfarb. The idea was to create an environment in which the several aspects of music representation could be considered and used. SMDL is an instance of HyTime and it inherits the functionality of this standard [9][23]. It divides the aspects of music representation in four domains: Logical (cantus), Visual, Gestual and Analytical.

The Logical Domain represents the intent of the composer when he writes a music piece. The Visual Domain incorporates graphic information and score notations. The Gestual Domain contains performance and executive information. The Analytical Domain embraces all music background information.

The Logical Domain (called *cantus*) codes music symbols in SGML syntax. Other domains use the hyperlink capabilities of HyTime to relate to other representation of specific music information [9].

The SGML syntax of SMDL is defined in the standard draft ISO 10743 [25]. The structure of information has been developed over a system of time-dependent axis called FCS (Finite Coordinate Space, an architectural form of HyTime) which is composed of cantus events (notes, rests, symbols). The other domains may refer to this representation to link the content of music

information.

MusicXML

The definition of this proposal [20] has been under development by Michael Good of Recordare. The content information is based on the MuseData and Humdrum [27] formats that explicitly represent symbolic music information in two dimensions (time/measure and part/instrument). Since XML has a hierarchical structure, the representation can be actualized as one of two conceptual hierarchies at a time. In the former representation, called Partwise, the sequence of measures is coded within each part. In the latter representation, called Timewise, the score is subdivided in measures, and subsequently each measure is subdivided in parts. Within each measure, or within each part, the symbols of the music events of the score are coded. An XSLT (eXtensible Stylesheet Language Transformations) [1] program has been developed that allows automatic translation between the two orderings. The MusicXML proposal is meant to represent a sufficient interchange format for notation, analysis, retrieval and performance application. The content represented by the format is score-oriented, e.g. the notes are represented as symbolic and graphical objects.

MusiXML

The aim of the proposal written by Gerd Castan [5] is to define a standard format for exchanging music score information. The structure is subdivided into three sections: Bibliographic Data Section, Logical Data Section and Filter Data Section. The last two sections are based on the idea of writing the minimal amount of redundancy information that separate content from style. The Logical Section is based on concepts of work-segments and Threads containing symbolic music content. The Filter Section is subdivided into *extracts* and *stoffs*. One extract is a grouping of stoffs with bibliographic data, brackets, braces and references to the stoffs. A staff contains references to threads and some other attributes. An XML DTD and an XML Schema [8] definition of MusicXML are available. Note that the MusicXML Schema is richer than the MusicXML DTD, because the former provides tools that the latter does not.

MusiCat & MDL

MusiCat, defined by Perry Roland [26], is a catalogue-oriented encoding format. A MusiCat document is organized as a catalogue document and has four main sections: Header, Body, Front and Back. The purpose of the description is to encode new and existing music catalogues, to favor popularization of electronic storage and transmission of music information. In fact, the schema doesn't provide elements to describe music content, it lets instead other notation formats represent it. Only one element (<notation>) containing text that

describes the music content is provided (in [18], the Humdrum format is referenced as an example).

MDL (Music Description Language) is the complementary definition for MusiCat to represent content music information in XML. It is a rich format with plenty of elements and attributes. The score is organized as the timewise structure of MusicXML. The attributes are subdivided in domains like those of SMDL (ie. Logical, Visual, Gestural and Analytical) and it gives the option to define new ones.

WEDELMUSIC format

It is the format for the representation of music information within the WEDELMUSIC project [36]. It is declared as to be an XML compliant format with multimedia capabilities, including identification, classification, symbolic, visual, versioning, printing, protection, image score, image, document, performance, video and lyric aspect. The underlying idea is to build an open format for the representation of WEDELMUSIC objects. It keeps separate the visual/formatting aspect and the symbolic one. In the description presented at the WEDELMUSIC conference [3], the structure and some examples about the identification, classification and music notation modelling are explained.

Musical Notation Markup Language (MNML)

Developed at IRDU (Internet Research and Development Unit) of the National University of Singapore, this proposal is an attempt to represent the entire music and lyric contents of a music score. MNML fully describes the basic melody and the lyrics of the piece. Capability for defining complex layout and other publishing features, performance annotations, and other embellishments are not considered [22][37]. A prototype of MNML editor and search engine has been developed.

Music Markup Language (MML)

It has been developed at University of Pretoria under the direction of Jacques Steyn. The development of the proposal is carried out with student projects. The aim is to define a markup language that can represent the structure of music and of music related processes. The authors have also developed some experimental software to graphically render an MML file [17].

MuTaTeD

MuTaTeD [21] stands for Music Tagging Type Definition. It is a project coordinated by Carola Boehm that intends to validate the concept of integrating the two existing music representation standards, SMDL and NIFF, and to provide a meta-standard for music mark-up (SMDL DTD)[25].

MusicML

Developed by Jeroen van Rotterdam, it is a simple draft that can represent only basic music information. The

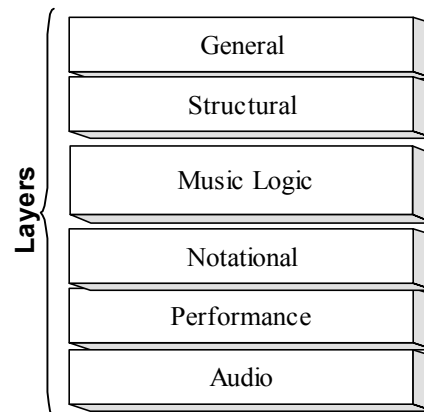


Figure 1. Structure of a Layered Symbolic Music Representation

author has also developed a Java applet to parse and visualize MusicML files. [19].

ChordML

This is a simple DTD developed by Gustavo Federico that can represent only lyrics and associated chords [6].

3 LAYERED SMI

SMI is organized in layers. Each layer models a different degree of abstraction of music information. An analogy can be made with the ISO/OSI layers architecture.

Essentially we distinguish among General Information, Structural Information, Music Logic Information, Graphic/Notational Information, Performance and Audio Information.

Music Logic Information is the core frame of our view of SMI. This information is composed by two fundamental concepts: the Spine and Logical Organized Symbols.

Spine is a structure that relates time and spatial information. With such a structure, it is possible to move from some point in a Notational instance to the relative point in a Performance or Audio instance. Logical Organized Symbols are the common ground for the music content. Music symbols are represented in the XML syntax making music content explicit to applications and users. Below we explain in details the concept of each layer.

General

The General Information layer is the place where metadata information about the music work is coded, such as:

- a) description of the music event (date, place)
- b) casting information
- c) the table of related music data files, referring to

all layers, with one or more files for the summarization of each layer

- d) the table of related multimedia data files, such as images, videos, and the like
- e) the table of related analog media
- f) technical information about related import/export/restoring/cataloguing/other operations
- g) general notes..

Any of the above mentioned items may contain all the related links to other layers.

Structural

The Structural information layer is about the explicit description of music objects and their relationships, from both the compositional and musicological point of view. There is no currently available standard which could be considered for this layer, but we can draw from:

- ANSI SMDL [25], which considers the topic, without giving any concrete tool for describing it
- High level music programming languages and formal tools for modeling [2], and particularly the descriptive approach based on Music Petri Nets [7][10][11][12][13].

The music information contents within this layer do not contain explicit descriptions of time ordering and absolute time instances of music events, while describing causal relationships among chunks of music information and related transformations and positioning within the music score, as they happen in the frame of compositional/decompositional - i.e. synthesis/analysis - processes.

Notational

Graphical and notational information regards the visual domain of music information. This layer embraces all the modalities in which the music can be written or read. This domain includes references to the content of images of scanned scores, and notational files in different formats such as NIFF and ENIGMA. Information contained in this layer is tied to the spatial part of Spine structure, allowing the localization of symbols.

Performance

This is the connection layer between symbolic performances and the SMI. Examples of source material considered in this context can be MIDI files, Csound or SASL/SAOL (MPEG4) compositions. Timed events can be suitably linked and synchronized to the SMI by means of the time part of Spine structure. Note that, in this context, time representation is expressed in relative coordinates.

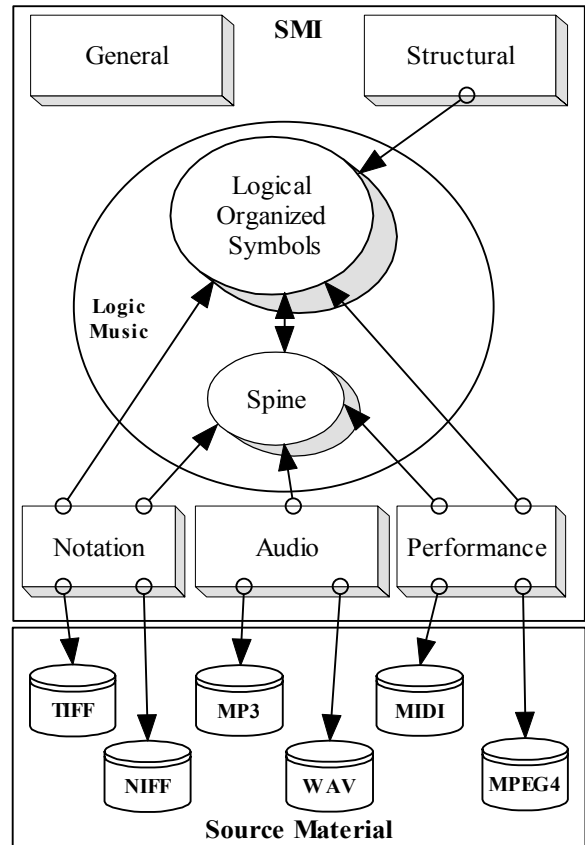


Figure 2: Relationships between SMI layers and between SMI and Source Material.

Audio

The Audio layer relates to Source Material that contains musical audio information. It is the lowest level of our view. In order to automatically relate it to the time part of the Spine structure, it is necessary to extract features indicating the actual temporization of musical events. Formats representing audio information can be subdivided in two categories: compressed or not compressed. Not compressed audio can be PCM/WAV, AIFF and Mu-Law. Compressed audio can be subdivided in lossy (e.g. MPEG and DOLBY) and lossless (ADPCM).

XML formats and layered SMI

Layered information is partially covered by some of the XML formats cited before. The most similar language is the WEDELMUSIC format that explicitly encode information on links to score images, notation, timed symbolic, audio and video files. The synchronization aspect is left to the application implementing the language.

MusicXML, MusiXML, MML and MDL represent symbolic information from different points of view. A common assumption is that the richness of the contents

must allow visual or aural rendering. They do not consider explicit integration with other kind of information such as audio, video, graphical or structural representation.

MuTaTeD and MNML are closed projects and the specifications are no longer available. MusicML and ChordML are very simple formats and they do not represent information about any other music aspect.

4 XML REPRESENTATION OF LAYERED SMI

Each layer described so far has a role in the context of the integration of the Music Work. XML organizes information in a hierarchical structure. Each layer is represented in a sub-tree of the Music Work. Relationships between layers (and thus between sub-trees) are implemented by means of XML references. Relationships between SMI and Source Material are implemented as descriptions of extracted parameters. As we can see in Fig.2, the central element of the representation is the Music Logic Information, to which all the other layers must refer. We will give a deeply description of this layer in the next paragraph.

The General information layer does not link directly to any layer, since it describes information about the whole Musical Work. Its XML representation is straightforward and does not need any particular description.

The Structural layer represents objects and relations between them. Objects are portion of the Music Logic Symbols layer. XML representation of a specific structural methodology is an open issue of this work.

The Notational Layer is the layer where visual information can be described. In our view, it should be possible to describe the presentation of the Music Logic Information in any media. Therefore, we split the structure of this layer in two sub-layers (or sub-trees): *layout description* and *linking information*. The former provides the tools to describe printing (e.g. A4 or Letter paper) or screen layout (e.g. desktop, palm or electronic book display). This description links directly to the Logical Organized Symbols and the Spine events. The latter provides descriptors for the inclusion and mapping of score images and binary notation file formats (NIFF, ENIGMA or even MIDI files) to the Music Logic Information layer. Furthermore, this is the place where information about security (watermark, signature, etc.) of each single graphic and notation file of Source Material can be inserted.

The Performance layer should contain generic information about the source material and the actualization information (such as synthetic instrument description, metronome indication and so on). Actualization parameters are the bridge between the time part of the Spine structure and the timed symbolic

information, since both represent timed information in relative coordinates. Thus, XML representation needs not be much detailed.

The Audio layer needs instead the representation of extracted features in order to be suitably integrated in SMI. And these features can be directly linked to the time part of the Spine structure.

5 MAX FRAMEWORK

MAX stands for Music Application using XML. As shown before, the core of our approach is the Music Logic Information view.

Spine

The Spine is a space-time structure composed of uniquely identified events. This is in fact the structure that relates Performance/Audio and Notational layers. Each event has two coordinates relative to the preceding event in its dimension, because some symbols can have meaning only in space or in time dimension. The first coordinate is the temporal coordinate of the event. It describes how much time must pass after the actualization of the preceding event. This time is expressed in Virtual Time Unit (**vtu**, as described in SMDL draft [25]). The second coordinate is expressed in a relative spatial unit that we call Virtual Logic Position Unit (**vlpu**). Its purpose is to represent the reference point for the vertical alignment of different symbols in different parts or staves.

A problem arising in the definition of such a framework is the discrepancy between Notation and Performance layers caused by repetition symbols. The question is: should the Spine structure be tied to the Performance or to the Notational layer?

If Spine refers to the Performance layer, it is necessary to create multiple references for symbols lying in repeated segments. On the contrary, if Spine refers to the Notational layer, a mapping from Performance/Audio to the Spine layer is needed. This is in fact a still open issue of our representation and a deep investigation is needed before a definitive choice can be taken. For the time being, we have chosen to be tied to the Performance/Audio layers.

Logical Organized Symbols

The root element of Logical Organized Symbols is a score. Since the rationale of our work is to separate the content from the presentation, we first distinguish between parts and staves. Different parts can be printed on the same staff, while one part can span different staves. Hence, the staves do not seem to be an essential part of the logical view. We represent them as a list with associated properties and identifiers. Each symbol is then linked to its own staff by means of an identifier reference.

Thus the part is the most important element of our view.

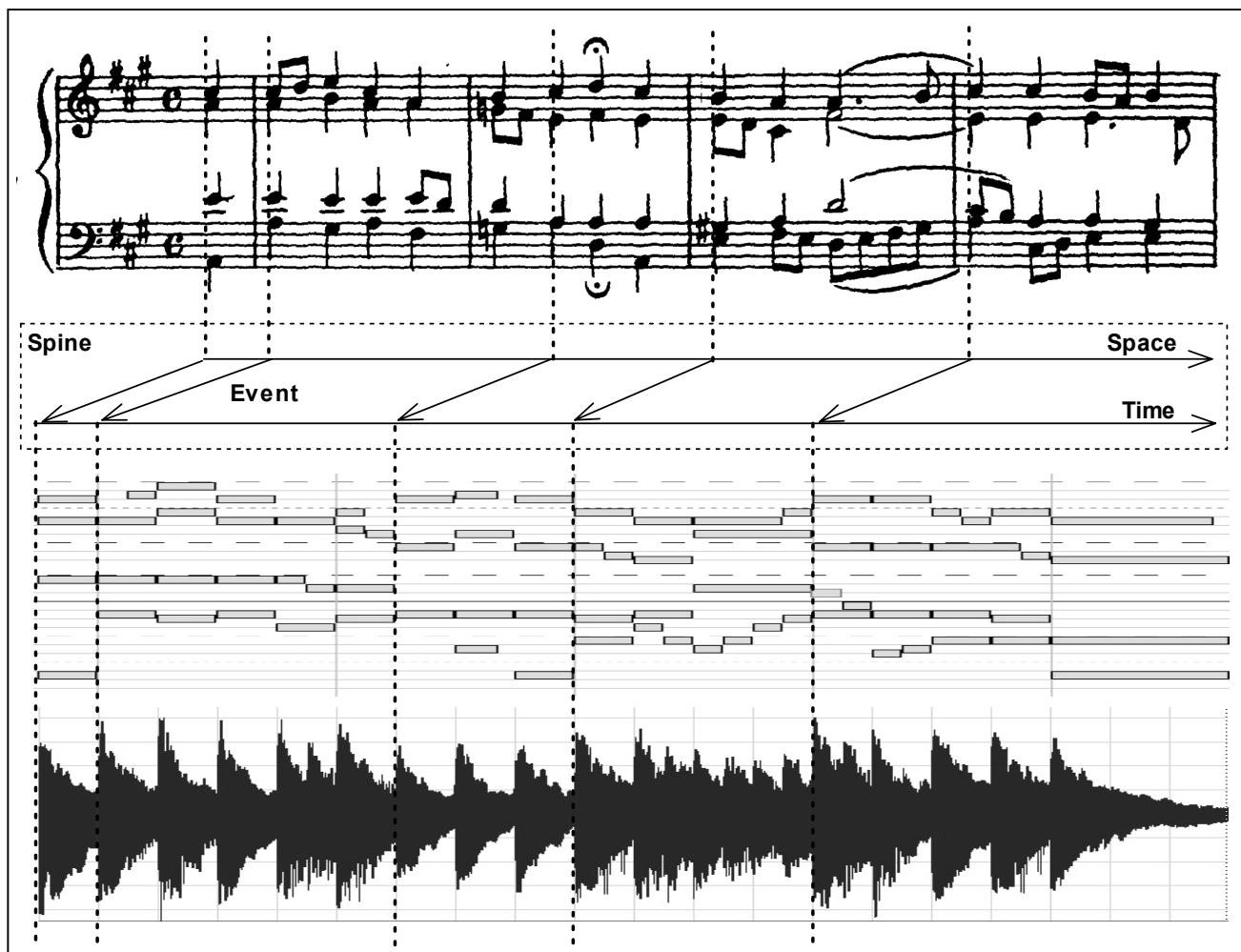


Figure 3 Spine. Relationships between Notational and Performance/Audio layer instances.

Each part is composed by a sequence of measures, within which the Notation Symbols are coded. Each symbol belongs to a voice that is referenced as a property.

Each Measure must have a reference number in order to be clearly identified and associated to the relative measures in other parts.

The music symbols can be rests or chords. As in NIFF [38], a single note is a particular case of a chord with only one notehead. Each chord and rest contains the link to the relative event in Spine, so that it can be localized spatially and temporally. Some other property can be also encoded about the logical description of the symbol. Note that the properties specified in this context have meanings only with respect to the definition of the symbol, because timing and localization information are managed by the Spine events.

The representation of beams, tuplets, dynamics, ties and other grouping symbols or embellishments can be efficiently managed as external elements that relate to

Music Symbols or to the Spine events. For example, the “crescendo” symbol must affect all the symbols present on a staff, and in this case it is much efficient to relate the symbol to the events in Spine, instead of linking to all the parts that relate to that staff.

There are several symbols that can affect the aural rendition of score, such as repetition bars, jumping signs and multiple endings. These structures can be summarized in XML with some elements that refer to segments in the Spine (since the repetition affect all the parts in the score). This formal tool defines projection of music symbols from the logical layer to the Performance/Audio layers. It avoids the encoding of redundant information and further separates the contents from presentation.

Lyrics can be thought as another logical component of our structure. The text is subdivided in syllables. Each syllable is tied to Spine and its relative part. This organization allows to overcome the problem of repetition

in the score while the lyrics changes and allows multiple lyrics to be encoded and extracted easily (e.g. for creation of librettos). It also simplifies the organization of text in layout rendering.

Layout

Logical Organized Symbols can be automatically rendered by the device-specific application, or can be specified by means of the Layout elements. The main container of this structure is the Layout, which is made by one or more Pages. On each Page, there can be one or more Systems or none. The System contains staves and lyrics at the same level. Systems are related indirectly to Logical Music Symbols through the Spine structure. Particular rendition of symbols and non-conventional signs can be easily modeled with SVG (Support Vector Graphics)[34], an XML standard for the definition of vector graphics.

Performance

In this layer, we describe properties to render aurally the music symbols and descriptors of Timed Symbolic Music Information (e.g. MIDI, SAOL, Csound). Rendering of music symbols can be performed by means of interpretative models that are depicted later in this article. Timed Symbolic information is naturally linked to the time part of a Spine structure, since event temporization is expressed in relative unit measures. Thus the descriptors must not encode the information about each event, but only general information about the instance and its relation to the SMI (e.g. the instance can be relative only to one instrument, or it may span only a part of the whole piece).

Audio

In this layer, information must be encoded that is extracted from the audio instance of the piece. Beside the usual generic data on the audio file, we think it is useful to encode actual timing parameters such as piece length, beat and localization of musical events. Moreover, this information is independent from the format in which the audio information is stored, since compressed formats are uncompressed before being played back.

6 RELATED TOPICS

The development of XML standards covers general topics that can be used in our context. We have identified some standards that could cover some of the topic of interest, such as security, layout symbol customization and versioning. These topics have also been addressed in the WEDEMUSIC format [36], to develop new methodologies to store this information.

Security

Ever since the widespread adoption of Internet as an exchanging medium, the topic of security has gained a great interest. In the context of our work, we distinguish

between the security encoded in the Source Material and the security expressed in the symbolic domain.

The Security in Source Material is out of the scope of this project. However, we face the problem of security in the symbolic domain. Here, we must further discriminate between security in XML (authentication SAML, access control XACML and digital rights management XRML) and security for XML (XML Signature and XML Encryption).

Security in XML is useful for the establishment of music computing environments, and of music services, based on XML. Security for XML is needed for the protection and ownerships validation of Music Works.

SAML

SAML (Security Assertion Markup Language) [30] is an XML-based security standard to exchange authentication and authorization information. It is being defined by the Security Services TC (SSTC) under OASIS. The working group envisions SAML as the exchange definition for authentication and authorization information.

XACML

XACML (eXtensible Access Control Markup Language) [31] is a core scheme for representing authorization and entitlement policies in XML. It is being defined by the XACML Technical Committee under OASIS. XACML is also expected to suggest a policy-authorization model to guide implementers of the authorization mechanism.

XRML

XrML (eXtensible rights Markup Language)[29] is a general-purpose, XML-based specification grammar for expressing rights and conditions associated with digital content, resources, and services. It is based on the rights expression language concept developed at the Xerox Palo Alto Research Center (PARC). XrML 2.0 expands the capabilities of a Digital Rights Language, usually seen in connection with the authorized use of protected digital content.

XML Signatures

The mission of this working group [32] is to develop an XML-compliant syntax for representing the signature of Web resources and portions of protocol messages (anything referenceable by a URI), as well as procedures for computing and verifying such signatures. This is a joint Working Group of the IETF and W3C.

XML Encryption

The mission of this Working Group (WG) [33] is to develop a process for encrypting/decrypting digital contents (including XML documents and portions thereof), and an XML syntax structure to represent the encrypted contents and information enabling an intended recipient to decrypt it.

Versioning

Personalization of scores is a common task for an orchestra director preparing the show. Often these modified versions have important value themselves, when made by important artists, and keeping tracks of these changes can be very complex. The most common are changes in music symbols, articulation, embellishments or indications written in any language. In our framework, this can be implemented simultaneously from the point of view of contents and graphical representation (particular signs can be modeled with SVG syntax).

7 OPEN PROBLEMS

The research in the area of computer applications in music has at least two fundamental open issues of interest for our conception of Musical Work: *Timbre* and *Interpretative Models*.

The timbre problem concerns how to automatically extract timbral features from complex audio signal, so that timbre modelling could be improved by means of automatic synthesis of timbral models and their parameters. For example, while we are quite able to get MPEG SASL codes that extract features from audio signals with constraints, there are no currently available coder which could concretely be used to get MPEG4 SAOL audio codes.

On the other side, we have the interpretative problem: how to translate NIFF codes (i.e. scores) into MIDI codes (i.e. performances) and viceversa? There is a need for a functional unit, currently not considered in the music information life cycle, that is, an Interpretative Models. In this way, for example, we can get timed, instantiated music information, coded in MIDI format, starting from score NIFF codes which contain only time-ordered music information. Vice versa, we could eventually have good automatic transcription from audio signals to music scores.

The development of such topics will be considered for the integration in the MAX framework to provide a complete high-level description of multilayered music information.

8 CONCLUSIONS

We have described the topics and concepts we intend to implement and develop in the context of the Recommended Practice for the Definition of a Commonly Acceptable Music Application Using the XML Language, carried out by XML Music Application Working Group (WG) [35].

Future work planned addresses mainly:

- integration of all available, meaningful experiences of other research groups working on XML & Music topic
- development of innovative topics of interest,

outlined in this paper; any cooperation is welcome.

As for all other XML standards, the main purpose of the MAX framework is to be extensible and open to the community of users of computer applications in music.

9 ACKNOWLEDGEMENTS

This work has been made possible by the effort of researchers and graduate students at LIM.

This research is partially supported under a grant by the Italian National Research Council (CNR) in the frame of the Finalized Project "Beni Culturali". The authors wish to acknowledge D. Baggi for improving the English language quality of the paper.

REFERENCES

1. AA. VV. Extensible Stylesheet Language (XSL) Version 1.0 *W3C Candidate Recommendation* 21 November 2000 <http://www.w3.org/TR/xsl/>
2. AA.VV. Standards in Computer Generated Music multiplatform mixed mode CD-ROM (Macintosh, Windows, Unix + CD-DA tracks), (G. Haus & I. Pighi Editors, 1996) IEEE Computer Society Press.
3. Bellini, P. Nesi, P. WEDELMUSIC Format: An XML Music Notation Format for Emerging Applications in *Proc. of WEDELMUSIC2001*, (Florence Italy, November 2001) IEEE Computer Society, 79-86.
4. Bray, T. Paoli, J. Sperberg-McQueen, C.M. Maler, E. Extensible Markup Language (XML) 1.0 (second Edition) <http://www.w3.org/TR/REC-xml>
5. Castan, G. Common music notation and Computers: Index. http://www.s-line.de/homepages/gerd_castan/compmus/index_e.html
6. ChordML Web site <http://www.cifranet.org/xml/ChordML.html>
7. De Matteis, A. Haus, G. Formalization of Generative Structures within Stravinsky's The Rite of Spring in *Journal of New Music Research*, 25,1 (Amsterdam, 1996) Swets & Zeitlinger B.V., 47-76.
8. Fallside, D. C. XML Schema Part 0: Primer. <http://www.w3.org/TR/xmlschema-0/>
9. Goldfarb, C.F. Newcomb, S.R. Kimber, W.E. Newcomb, P.J. Information Processing – Hypermedia/Time-based Structuring Language (HyTime) - 2nd edition. *ISO/IEC JTC 1/SC 18 WG8 N1920rev*, May 1997.
10. Haus, G. Sametti, A. SCORESYNTH: a System for the Synthesis of Music Scores based on Petri Nets and a Music Algebra *IEEE Computer* 24, 7 (July 1991) 56-60.
11. Haus, G. Rodriguez, A. Formal Music Representation; a Case Study: the Model of Ravel's Bolero by Petri Nets in "Music Processing in *Computer Music and Digital Audio Series*, (Madison WI, 1993) G. Haus Editor, A-R Editions, 165-232.

12. Haus, G. Sametti, A. Modelling and Generating Musical Scores by Petri Nets in *Languages of Design*, 2,1 (Amsterdam, 1994) Elsevier Publ., 7-24.
13. Haus, G. Sametti, A. Score Analysis/Resynthesis Environment of the Intelligent Music Workstation in *Journal of New Music Research*, 24,3 (Amsterdam, 1995) Swets & Zeitlinger B.V., 230-246.
14. Haus, G. Rescuing La Scala's Audio Archives. *IEEE Computer*, 31(3), pp.88-89, IEEE CS Press, Washington, 1998.
15. Haus, G. Longari, M. Coding Music Information within a Multimedia Database by an Integrated Description Environment. *Atti del XII° Colloquio di Informatica Musicale*, AIMI/Università di Udine, Gorizia, 1998
16. Haus, G. Pelegrin, M. L. Music Processing Technologies for rescuing Music Archives at Teatro alla Scala and Bolshoi Theatre in *Journal of New Music Research*, (Amsterdam, 2001) Swets & Zeilinger.
17. MML Web site <http://www.mmlxml.org>
18. MusiCat Web site
<http://icarus.lib.virginia.edu/~pdr4h/musicat/>
19. MusicML DTD site link
<http://www.oasis-open.org/cover/musicML-DTD.txt>
20. MusicXML DTD web site
<http://www.recordare.com/xml.html>
21. MuTaTeD
<http://www.pads.ahds.ac.uk/mutated>
22. Narayan, K. Chiam, P. Wei, Y. Kok, L. Tin Wee Y. and T. Delivering Music Notation over the Internet: Syntax, Parser, Viewer and GUI-Input System. *Proc. of Inet'96*, <http://irdu.nus.edu.sg/music/>
23. Newcomb, S.R. Kipp, N.A. Newcomb, V.T. The HyTime. Hypermedia/Time-based Document Structuring Language. *Communication of the ACM* 34:11, November 1991.
24. Newcomb, S.R. Standard Music Description Language complies with hypermedia standard. *COMPUTER*, vol. X N.Y. 24:7, pp. 76-79, July 1991
25. Newcomb, S.R. Standard Music Description Language. ISO/IEC DIS 10743
<ftp://ftp.ornl.gov/pub/sgml/WG8/SMDL/10743.ps>, June 1995.
26. Roland, P. Extensible XML4MIR: Markup Language for Music Information Retrieval. *Proc. of Music IR 2000*
27. Selfridge-Field, E. Beyond MIDI The Handbook of Musical Codes. *The MIT Press*, Cambridge, Massachusetts, 1997
28. Sloan, D. Aspect of Music Representation in HyTime/SMDL. *Computer Music Journal*, Cambridge, MIT Press 17:4, pp 51-59, Winter 1993
29. <http://www.xrml.org/>
30. <http://www.oasis-open.org/committees/security/>
31. <http://www.oasis-open.org/committees/xacml/>
32. <http://www.w3.org/Signature/>
33. <http://www.w3.org/Encryption/2001/>
34. <http://www.w3.org/TR/SVG/>
35. <http://www.lim.dsi.unimi.it/IEEE/XML.htm>
36. <http://www.wedelmusic.org>
37. <http://www.oasis-open.org/cover/mnml199906.html>
38. <http://www.musique.umontreal.ca/personnel/Belkin/NIFF.doc.html>
39. <http://www.lim.dsi.unimi.it>