Modeling Music Notation in the Internet Multimedia Age

P. Bellini, P. Nesi

Department of Systems and Informatics, University of Florence, Florence, Italy
Fax: +39-055-4796425, tel: +39-055-4796363
Corresponding author: nesi@dsi.unifi.it
see also for the research group: http://www.dsi.unifi.it/~nesi/

Abstract

Music notation modeling is entering in the new multimedia internet age. In this era new interactive applications are appearing on the market, such as software tools for music tuition and distance learning, for showing historical perspective of music pieces, for musical content fruition in libraries, etc. For these innovative applications several aspects have to be integrated with the model of music notation and several new functionalities have to be implemented such as: automatic formatting, music notation navigation, synchronization of music notation with real audio, etc. In this chapter, the WEDELMUSIC XML format for multimedia music application of music notation is presented. In addition, a comparison of this new model with the most important and emerging models is reported. The taxonomy used can be useful for assessing and comparing suitability of music notation models and format for their adoption in new emerging applications and for their usage in classical music editors.

KeyWords: music notation, multimedia music, digital right management, MIDI, WEDELMUSIC, XML, music notation formatting, multimedia integration.

1 - Introduction

The modeling of music notation is an extremely complex problem. Music notation is a complex piece of information which may be used for several different purposes: audio coding, music sheet production, music teaching, entertainment, music analysis, content query, etc. In the Internet Multimedia age music notation and associated applications are being left behind. Many other applications are getting the market and most of them will become more diffuse in a short time. A unique and comprehensive format for music incorporating other media is required. In order to satisfy diverse applications several aspects have to be considered, ranging from problems of information modeling to integration of the music notation itself. The main problems are related to the organization of music notation symbols in a suitable and acceptable form. Such a set of rules has never been formalized, although they are informally used in several books and considered part of the experience of professional copy editors (engravers) of music sheets. Other problems include adoption of classical music notation in applications subject to DRM (digital right management), versioning, multimedia integration and navigation, etc.

The evolution of information technology has recently produced changes in the usage of music notation, transforming the visual language of music from a simple coding model for music sheet, to a tool for modeling music in computer programs. Enabling cooperative work on music and other information integration tasks is also necessary. More recently, users have discovered the multimedia experience, and thus, the traditional music notation model is likely to be replaced with something much more suitable for multimedia representation of music. The improved capabilities of computer programs are going to solve also early music notation formatting problems with powerful tools based on artificial intelligence technologies.

In this chapter, the main problems and the most promising evolution trends of music notation are reported. To this end, a comparison of the most representative music notation formats is included. The comparison has been performed on the basis of a set of aspects which has been related to major problems mentioned in the first part of the chapter.

The Chapter also introduces the WEDELMUSIC (WEB Delivering of Music) (Bellini et al., 2001) (Bellini et al., 2003) model and language. WEDELMUSIC does not claim to solve all problems related to music notation modeling, but it provides an effective framework that includes a comprehensive music notation and many innovative aspects suitable for multimedia and future applications. On such basis several problems highlighted in Byrd (1984) are solved. Other object-oriented music models (e.g., (Taube, 1998), (Pope, 1991)), do not model relationships among symbols with the standard of accuracy required. They adopt a different object-oriented model, since they are mainly focussed on producing sounds instead of visually presenting music to musicians. The WEDELMUSIC model allows the organization of rules for positioning notation elements on the staff, while considering their multiple representations and giving the possibility of including instrumental symbols and enabling their distribution in the network. In addition, WEDELMUSIC is a version of MOODS (Music Object Oriented Distributed System) a cooperative model for editing music notation (Bellini, Nesi and Spinu, 2002).

2 - Background

The problems of notation modeling and visualization of music scores have been addressed in computer systems several times -- (Anderson and Kuivila, 1991), (Blostein and Haken, 1991), (Dannenberg, 1993), (Dannenberg, 1990), (Rader, 1996), (Gourlay, 1986), (SelfridgeField, 1997), (Blostein and Baird, 1992), and (Pennycook, 1985).

Among many possible computer-based applications of music, the notation editing for professional publishing and visualization is one of the most complex for the intrinsic complexity of music notation itself (Ross, 1987), (Heussenstamm, 1987), (Wood, 1989), (Byrd, 1984). Music publishing requires the production of high quality music scores in terms of number of symbols and their precise placement on the staff. Music scores have to be presented to professional users (musicians) with specific relationships and proportions among symbols. The formal relationships among symbols are often neglected or misused by most commercial computer-based editors. Their aims consist of producing music by means of MIDI (Music Instrument Digital Interface) interfaces or sound-boards, or in acquiring music by means of a MIDI interface, or simply in editing music on the computer so as to manage sound-boards, etc. In such events, a limited number of music notation symbols is needed (e.g., notes, rests, accidentals, clefs and points -- approximately 50 symbols), which makes impossible to reproduce through a MIDI interface the different effects specified by many others music notation symbols.

Sometimes the sets of music symbols available in professional editors (e.g., Score, Finale, Sibelius, etc.) for publishing music include also *instrumental and execution symbols* (symbols for describing the behavior of the musicians while playing a specific instrument -- e.g., up or down bow, with mute, without mute, fingering, string numbers, accents, etc.) for: strings, harps, drums, flutes, etc.; *orchestral and repeat symbols*, etc. These symbols together with many others, are needed when main scores and parts for classical music, operas and ballets have to be produced, and they are mandatory when scores are used in music schools to train students to perform specific executions and interpretations of music. The number of elementary symbols is close to 300 in the commercial editors which are commonly used to prepare music scores for publishing, Such symbols are frequently treated as components to implement more complex symbols. Typically, musicians have to personalize/prepare a score for the execution; therefore, even this great number of symbols is not powerful enough to meet all their needs. On these grounds, some music editors provide a font editor to help the user in adding/creating symbols; which unfortunately, are treated as simple graphic entities.

Commercial music editors for publishing are mainly inclined to place music symbols on the score page rather than to collect relationships among symbols and then organize the visual information accordingly. They are orientated towards printing music, since this is what they consider their most important application. They provide a complete set of symbols which a user skilled enough in both music notation and computer graphic interface can use in order to produce professional music sheets. In music editors of this kind, the arrangement of many music notation symbols is often left to the users' competence. Music symbols (excluding notes, rests and a very few other symbols) are mainly considered by music editors as graphic elements to be placed in any position on a score page without feeling bound to address any problems related to the music notation like, for instance, the visual and syntactic relationships among symbols. This results in many music editors allowing several notation symbols to be placed in incorrect positions, thus producing strange and incorrect music scores. Eventually it follows that users are left with no technical support on how to place symbols.

For these reasons, professional music editors are powerful, but at the same time they are difficult to handle for non-expert users in music notation. Typically, musicians can read music, but they are pretty unfamiliar with rules for arranging symbols (e.g., when a symbol is associated with symbol B, symbol C has to be moved up one-half unit). Musicians have no problem when asked to read a correctly annotated score, but can be puzzled when they are faced with non-perfect visual constructs. Musicians are in fact artists, not music engravers nor notation technicians. It often occurs that conductors and composers are more sensitive to problems of music notation and visual expressiveness; while archivists in music theaters, music engravers, and music notation teachers are the experts of music notation problems.

2.1 - Emerging New Applications

Recently, with the spread of computer technology into the artistic fields, new needs for computer-based applications of music have been identified: (i) cooperative music editing in orchestras and music schools, as with the project MOODS ESPRIT (Bellini, Fioravanti and Nesi, 1999) (Bellini, Nesi, Spinu, 2002); (ii) music score distribution via Internet, as with many WWW sites distributing music scores or MIDI files; (iii) multimedia music for music tuition systems: edutainment and infotainment. We have started to investigate these two new fields since 1994. MOODS consists of an integrated system of computer-based lecterns/stands for cooperative editing and visualization of music. MOODS is an innovative solution for automating and managing large amount of information used by (i) orchestras during rehearsals and public performance of concerts, operas, ballets, etc. (ii) students of music during lessons in conservatories and music schools, (iii) publishers during massive editing of music.

The targeted MOODS end-users are theatres, itinerant orchestras, groups of musicians, music schools, television network orchestras, and publishers of scores. MOODS can be used to: (i) reduce the time needed for modifying main scores and parts during rehearsals in a cooperative way; (ii) manage (load, modify, and save) instrumental and personal symbols on main scores and parts; (iii) manage and reproduce the exact execution rate at which each measure of a score has been performed; (iv) automate page turning during rehearsals and final performances; (v) change music pieces quickly or restart from marked points; and (vi) manipulate the main score and all instrument parts together with the full music score in real time. Computerized music lecterns can be used by musicians to avoid transporting heavy paper music scores, to save their work, to manage versions, etc. A distributed system with the above features must be capable of showing music in different formats and at the same time must support cooperative editing of music in different formats, which means showing the changes of one operator to the other ones in real time. A public demonstration of MOODS functionalities was given at La Scala Theatre in Milan. The main difference between classical music editor and MOODS is the availability of cooperative work on music and the presence of integrated multimedia aspects. When cooperative work is relevant and the music has to be visualized at several resolutions regardless of the visualization support features -- for example, low/high resolution monitors and printers -- the following two main functionalities have to be available.

1. a clear distinction between the music notation model and the visualization rules: automatic reformatting while taking into account the user's needs, transposition, page dimension, etc.
2. music notation model has to be abstract enough to allow the *interactivity with music at a symbolic level*: adding and deleting symbols, changing symbols features and making those changes available regardless of the visualization details without any reloading of the music score. To this end, the music notation and model has to support an indexing model (Bellini, Nesi, Spinu, 2002). This makes it possible to define policies of versioning, selective and

un-linear undo, navigation among music notation symbols, etc.

This feature seems to be very far from being a reality in present Internet applications, even though it is of great interest for cooperative applications -- e.g., virtual orchestras. It is noteworthy thinking of cooperative applications as something which will undergo a strong implementation among Internet Applications in the very near future.

For the second type of application music can be distributed by using images, audio and symbolic files. At present, the distribution of music via Internet consists only of images of music scores or simple symbolic files, while audio files are widespread and better distributed. The music received from Internet can be either *interactive or not*. For interactive music we intend music that can be manipulated in a certain measure: addition/deletion of symbols, transposition, reformatting, etc., without the violation of the copyright law. Images of music sheets do not allow the music manipulation, while the MIDI model is too coarse to meet the needs of professionals, because MIDI provides a reduced set of music notation symbols.

In the past, the language called NIFF (Notation Interchange File Format) was supported by editing/publishing and scanning programs, for creating an interchange format for music notation (NIFF, 1995). In the following years, the NIFF format has been abandoned as an interchange format by several companies and publishers. Recently MusicXML has been proposed with the same aim (Good, 2001).

The Internet is at present dominated by the distribution of documents through the so-called mark-up languages derived from SGML, HyTime, XML, etc. (Sloan, 1997). A mark-up language consists of a set of constructs to express how text has to be processed with the main aim of text visualization. Generalized mark-up languages specify only what you have, rather than how to visualize it. The visual aspects are specified by using standard tags which state the typographic features: spacing, font change, etc. For example, when using XSL a formatting style should be defined. This means that a clear distinction has to be made between the content and the formatting aspects. These languages are mainly conceived for describing monodimensional pieces of information like texts and fail to model relationships among symbols at a logical level. For this reason, mark-up languages do not fit for any direct support to cooperative work; since they are not structured enough to be independent of the formatting aspects. In fact, mark-up languages were created for formatting textual documents and not for defining relationships among document symbols. This is one of the main problems which prevents from an unreserved adoption /acceptance of SMDL or other mark-up languages. Recently, several other XML compliant mark-up languages for music modeling have been proposed, among them: MNML (Musical Notation Markup Language), MusicML, MML (Music Markup Language), MusicXML, etc. The MNML is a simplified version of SMDL. In MNML, it is possible to describe completely the basic melody and the lyrics of a music piece; on the other hand it fails to describe all the needed details of real music scores (annotations, publishing aspects, etc.). MusicML can only represent notes and rests. Even staccato points and slurs are missing.

The adoption of a symbolic model totally separate from the visualization issues makes the distribution of interactive music a very complex task. In MOODS (Bellini et al., 1999), a cooperative editor of music has been proposed. It was built on a non-XML-based format. The solution was the definition of a specific language called MILLA (Music Intelligent Formatting Language) able to define rules for a formatting engine while letting users also have the possibility of forcing exceptional conditions. WEDELMUSIC model and language can be considered the XML evolution of MOODS format. With WEDELMUSIC several MOODS' early problems for music modeling have been solved, in addition WEDELMUSIC is a multimedia model. WEDELMUSIC provides an effective framework which includes most music symbols and their relationships. On such basis several new and innovative applications can be built and some exceptions and several modeling problems already highlighted in Selfridge-Field (1997), Gourlay (1986). WEDELMUSIC format presents multimedia capabilities and includes identification, classification, symbolic, visual, protection, animations, image score, image, document, performance, video, lyric, aspects, synchronization, etc. It keeps separate visual/formatting from symbolic aspects. WEDELMUSIC format can be profitably used for new applications and as a format for interchanging symbolic description of music scores. Formatting rules and the corresponding MILLA engine of WEDELMUSIC can be found in Bellini, Della Santa and Nesi (2001).

The distribution of music sheets via Internet can be considered one of the emerging new applications of music notation. Presently great part of the music sheets distributed via Internet is in the form of PDF or Image files. In both cases, the music delivered is not interactive since no transposition, or annotation can be performed. Other more interactive solutions have been produced by Coda/Finale, Sibelius and Noteheads. They distribute music notation in their proprietary formats and this can be viewed and played in an Internet Browser via MIDI by using specific plug-ins or ActiveX. For these applications, the Digital Right Management, DRM, capabilities are particularly relevant to protect the distributed content. Typically, the models permit printing the music sheet only a limited number of times. In this field, WEDELMUSIC propose an integrated solution for distributing multimedia content, files in any format, including also music score in a protected manner among archives and these to their attendees (Bellini, et al. 2003).

2.2 - Music Notation Problems vs new Applications

This section deals with the most important modeling problems adopting the perspective of the emerging applications of music notation. As stated by several authors in the past, the modeling of music notation presents several problems: (i) the intrinsic complexity of formalizing both *music notation and the relationships* among music symbols; (ii) the necessity of providing different visualizations/views of the same music as it occurs with main score and parts; (iii) the complexity of any automatic formatting of music symbols; (iv) the necessity of adding new symbols in order to expand the music notation model and make it fit for modern music and users' needs; (v) the necessity of presenting new functionalities for multimedia applications; (vi) the necessity of producing high quality music sheets in terms of tiny adjustments of symbols so as to avoid collisions and produce very clearly written and well recognizable music sheets at a first glance, (vii) the formatting of music around the page end.

The modeling of all possible relationships among notation symbols is a complex task. As the number of symbols grows, the number of relationships among them grows with a ration more than proportional. The syntax and semantics of music are strictly related and cannot simply be modeled by using only *non-visual* (relationships among symbols) or *visual* aspects of music (positions with respect to the staff). The modeling of music symbol **relationships** is not enough to describe the music notation: for example, a *diminuendo* starting between two notes (and ending on the second) depicts the instant in which it has to be applied. The description of music as a collection of **graphical symbols** on the page regardless of their relationships relies too much on the visualization device (automatic reformatting is really complex). In most commercial music editors, the music notation symbols are simply managed as graphical symbols that can be placed on the screen.

The automatic generation of the main score from the different parts and the extraction of parts from a main score are complex tasks for real music, since the conductor's main score is typically formatted in a different way from the musicians' scores. Parts are obtained by using compressed versions of music notation symbols, and also present are several *instrumental, personal symbols*, which are typically omitted in the main score. The main score may present some specific textual notes or symbols for the conductor. The main score may show more than one part on a unique staff, in this case, parts are treated as distinct voices/layers. Other indications are added in the parts, so as to make the coordination among musicians easier; for instance, the so-called grace/cue notes. The main score and parts present different justifications, line breaking and page breaking. In some music editors, even the insertion of a measure in a part results in a huge manual work of reformatting in order to provide the replication of the performed changes on the main score, or vice versa. In

Dannenberg (1990), a solution was given, suggesting the adoption of a unique model and distinct views for showing music on the basis of each specific need and context.

The positioning of music elements on the score presents many critical conditions. A music editor should help the users to produce well-formatted music scores. In most of the music editors, this task is addressed by considering only the visual representation of music symbols and not their relationships. As above highlighted, some visual aspects of music scores cannot be formalized in terms of symbol relationships. For this reason, the visual arrangement of music notation symbols makes the description of the music scores complete. Music can be considered a visual language with its own rules. On this ground, the modeling of visual rules for formatting music is a requisite step to the completion of the model (Bellini, Della Santa and Nesi, 2001).

Typically, there is the need of adding new symbols for stating specific instrument aspects, and/or expanding the music notation model to take in modern music as well. The addition of new symbols means to add fonts, relationships among other symbols, execution rules and formatting rules. For example, the addition of a symbol modeling a special figure (such as a cloud of notes) implies a deep change in the music notation model; whereas the addition of a new marker for notes can be performed by adding positioning rules and fonts, since the relationship with the note is already established. Several other symbols must be added to fulfill the needs of all instruments of a large orchestra (specific accents, jargon symbols, heel, toe, bridge, etc.). Most of them put on different visual representations depending on the instrument they are used for.

At a first glance, the new applications related to Internet distribution of music scores seem to be not too much different from the music editors currently on the market. The applications of the next few years will be mainly based on: (i) cooperative work on music notation, (ii) interactivity of music notation in the respect of the owner rights, (iii) the availability of different integrated information associated with the music piece, (iv) music as support for tuition systems. In all these instances, music has to be independent from any visualization support features, formatting features and resolution aspects. Then, the following functionalities have to be available.

- a clear distinction between the music notation model and the visualization rules: automatic reformatting on the account of the user's needs, etc.
- music notation model has to be abstract enough to allow interactivity with music at the symbolic level: adding and deleting symbols, changing symbols features, selective and unlinear undo, versioning, etc.
- music model has to integrate several aspects: audio, symbolic music notation, images of music score, documents, video, animations, www pages, multilingual lyric, etc.
- mechanisms for distributing music by using shops, libraries, conservatories, etc., as local distributors.
- a refined protection model for Digital Right Management.

The first two problems may be managed by formal models supported by a separate engine for the automatic formatting of music according to precise rules. Ideally, this work is infeasible (Bellini, Della Santa, Nesi, 2001), but positive and promising compromises can be obtained. The new multimedia applications are bringing music in a new era. Simple audio files or music sheets are undergoing important changes in order to be included in more complex multimedia objects.

3 – Music Notation Models and Languages

During our work for defining MOODS and WEDELMUSIC models and languages several other music notation languages and tools were reviewed and analyzed; among them: MIDI, Finale, Score, MusiXTEX, NIFF, MusicXML, and SMDL. In this section, a short discussion and comparison among these formats is reported. The comparison does not claim to be exhaustive; it is focussed on the aspects relevant for the adoption of these languages and models in the new applications. Please note that WEDELMUSIC model and format is discussed in the next section and thus its description is not summarized in this section.

MIDI, Music Instrument Digital Interface

MIDI is a sound oriented language. It is unsuitable for modeling the relationships among symbols and for coding professional music scores: in MIDI accents, ornaments, slurs, etc. are missing or very difficult to recognize. MIDI is the most widespread coding format among those that can be found on Internet since (i) it can be easily generated by keyboards, (ii) it has been adopted by the electronic music industry for representing sounds in a computers form, (iii) MIDI music can be played and manipulated with different synthesizers. A very large percentage of these files have been generated without the intention of recovering the music score and therefore they are do not fit as music scores due to the presence of several small rests or notes. There is also a limited space for customizing the format. This space has been used for defining several versions of enhanced MIDI formats (SelfridgeField, 1997). In the literature several extensions of MIDI have been proposed but no one has really got the large diffusion to substitute the classical MIDI format. MIDI format is mainly used as an interchange format, on the other hand, its capability in modeling music notation is very limited. Most of the music editors are capable of loading and saving music notation in MIDI.

SCORE

SCORE is probably the most widespread notation editor among publishers for the high quality and professional Postscript printing (Smith, 1997). In SCORE each symbol can be precisely arranged on the page according the engravers' needs. Complex symbols can be produced by using graphic elements on the music sheet. Several minor accompanying symbols of the note can be placed on the staff in any position and thus the relationships with the note are non-defined. This means that a movement of the note or its deletion does not influence the life of these symbols. SCORE presents no distinction between slur and ties and no automatic conversion of sequences of rests into generic rests. SCORE is a page-oriented editor, in the sense that music information is collected only related to the preparing page: the editor is capable of managing only a printable page at time. Since the music is created page by page, parts (each page a file), are very hard to automatically extract because the match is based on part numbering and the numbering of staffs may be different in successive pages (some parts can be missing, the numbering is not ordered from top to bottom). When symbols are managed in the middle of a line or page break, they are drawn with two distinct graphic lines. This makes complex the extraction of parts and the reorganization of music measures along the main score and parts. The insertion of some measures in a page or the changing of the page dimensions may need the manual manipulation of all pages. This is totally unacceptable for music which has to be distributed via Internet and which has to be viewable on several different devices for both size and resolution, with changes to be performed in real-time.

MusiXTEX

MusiXTEX is a set of macros for LaTEX and/or TEX for producing music sheets (Taupin, Mitchell, Egler, 1997), (Icking, 1997). The language is interesting since its structure is substantially symbolic while graphic commands can be added to provide precise positioning. The relationships among symbols depend on the order symbols appear in the code. The language is printer-oriented and thus it allows the placement of graphics symbols anywhere on the page.

Some simple rules for the insertion of symbols are available (definition of up and down of the stems for notes). With MusiXTEX specific rules for the visual organization of symbols on the page could be defined exploiting the power of LaTEX and TEX. Classification features could be implemented by using a similar technique. In the past we developed a simple music editor based on an early version of MusiXTEX, it was called LIOO, Lectern Interactive Object Oriented. It has been the early version of MOODS system.

In MusiXTEX, the work has to be manually performed during the score coding. MusiXTEX does not support: (i) the automatic beaming (identification of the groups of notes to be beamed together in the measure), (ii) the automatic definition of stem direction of notes in beams, (iii) the automatic management of positioning for accents.

NIFF, Notation Interchange File Format

NIFF was developed with the perspective of defining an interchange format for music notation among music notation editing/publishing and scanning programs (NIFF6, 1995). Its present version if 6b. NIFF was derived from design rules of the Resource Interchange File Format (RIFF). NIFF design resulted from a quite large group of commercial music software developers, researchers, publishers and professionals. For this reason, the model was defined with the aim of supporting the most relevant aspects of several models. This represented a great limit in the expressiveness of the languages and models for describing exceptions. The main features of NIFF are: (i) a feature set based on SCORE, (ii) division of visual information in page and non-page layout information, (iii) extensible, flexible and compact design, and (iv) inclusion of MIDI data and format. Since 1995, the notation has not been improved. It is currently supported by LIME editor and few others. Since 1996, a kit for implementing a loading module for NIFF files is available.

NIFF language includes in a unique model both visual and logical aspects. This makes it difficult to deliver music regardless of the visualization details, which is a feature needed in cooperative applications. Relationships among symbols are defined via specific links (anchorage). They can be set according to default rules but specific actions can be performed for managing some changes in the editors. In NIFF, no support is provided for either versioning or cooperative work, since each logical element in the format cannot be univocally identified. NIFF presents a limited number of symbols but grants the possibility of defining new symbols.

SMDL, Standard Music Description Language

SMDL is a mark-up language built on SGML and HyTime standards (SMDL10743, 1995). The aim was the definition of an interchange abstract model. SMDL model includes the following aspects: logical, gestural, visual and analytical. The logical aspect includes the music content (pitches, rhythms, dynamics, articulations etc.), that is, the abstract information common to both gestural and visual domains. The gestural domain describes specific performances of the logical domain. It includes dynamic symbols etc. and MIDI information. In MOODS, these aspects are collected in logical parts. The different versions due to the gestural aspects can be managed in MOODS via the versioning mechanism.

In SMDL, the visual domain describes the musical typographic details of scores (the symbol, placing, fonts etc.). This is quite similar to the concept of MOODS but SMDL does not include mechanisms for defining visualization rules for symbols. The analytical domain consists of music analyses for classification purpose and a support for defining more sophisticated analysis in the logical and gestural parts. SMDL was analyzed in depth in the CANTATE project (CANTATE, 1994). The result of the analysis was as follows: SMDL cannot be used for modeling scores. It can only produce visually visible scores by using other formats such as FINALE, SCORE, NIFF, etc. or images of music sheets. SMDL cannot be used as a standard interchange format for the visual aspect but only for the logical aspects of music: a sort of container in which several different music formats and related information can be collected and organized (NIFF, MIDI, animations, textual descriptions, etc.) Currently there is no commercial SMDL software for editing music or producing music digital objects. In the CANTATE project an application of SMDL for the distribution of music for libraries was developed.

SMDL presents a neat distinction between the visual and the logical parts that should be modeled by different languages and formalisms. For this lack of integration among music aspects, SMDL cannot be used for either distributing interactive music via Internet or as a basis of cooperative work on music scores like in MOODS. More recently, several other mark-up languages for music modeling have been proposed. Unfortunately, they are weakened by the lack of managing slurs, accents, etc. The adoption of a structural model totally separate from the visualization issues makes the production of professional music a very complex task. In MOODS, the solution was the definition of MILLA meaning a specific language for defining visualization rules and giving users the possibility of forcing exceptional conditions.

MusicXML

MusicXML is a XML format for music notation interchange developed by Recordare (Good, 2001). It is based on two other textual formats for music notation representation, the MuseData and the Humdrum formats (Selfridge-Field, 1997). It represents the music in a time-wise (parts nested within measures) or part-wise (measures nested within parts) manner, a XSLT allows the transformation from one to the other format. The format covers the western musical notation from 17th century onward, it is mainly oriented to describing the logical structure of music even if some graphical detail could be added. A plug-in for Finale allows it to load and save files using this format, the Sharpeye OMR application uses it as a interchange format with Finale. It includes a subset of the most commonly used formats. It can be useful for interchanging music notation but is far from being a good format for new multimedia applications. At the XML level, MusicXML is strongly based on Tag rather than on Attributes. This limits the flexibility of defining new symbols that are considered values of attributes.. The addition of a different value is simpler that the production of a new rule for managing a new tag of the XML

FINALE, Enigma format

Music produced by FINALE program is coded in Enigma format. This format is partially documented. The editor and format are mainly oriented towards page preparation rather than defining relationships among symbols. Clear evidence of this is found in several symbols being non linked to figures and simply placed on the page, assuming any position without any bound to follow the sort of the note whenever moving or deleting. The format has been recently improved and allows the definition of some relationships among music notation symbols. This is not its internal philosophy but it is left to the users. The Finale model does not present a clear trace for voices that pass from one staff to the other in multi-staff parts, such as those for Piano, Harp and Organ. In several cases, the arrangement of music notation symbols is quite hard since the automatic mechanism for completing the measure is quite disturbing.

GUIDIO format and tools

It is a textual format (human readable) for symbolic music description. The description is extremely compact and it seems to be optimized for direct user entry rather than to have an editor to produce it (like MusiXTex). A set of tools are present to transform this description to MIDI and to render it as PostScript or GIF, or to convert to it MIDI and FINALE files to GUIDO. GUIDO language has been designed in three layers, Basic GUIDO describes the basic musical symbols of western music notation (notes, rests, slurs, etc.) and their structure (staffs, voices, chords), Advanced GUIDO extends the Basic GUIDO to support exact score formatting and more sophisticated musical concepts, Extended GUIDO introduce concepts beyond conventional music notation. The tools currently support Basic GUIDO and Advanced GUIDO. However the specification of Advanced GUIDO and Extended GUIDO are not available. Automatic formatting rules are encoded in the renderer and automatic beaming are supported, however precise positioning and beaming can be specified forcing the position. The symbols supported in the Basic GUIDO are few and cover only the most important expressive indication (staccato, tenuto, accento, marcato…) and ornaments (trillo, mordente, gruppetto, tremolo, glissando) and no symbols for specific instrument are present (violin, piano, arpa, etc.). Moreover GUIDO lacks in the possibility to introduce new user defined symbols.

4 - The basic aspects of WEDELMUSIC Model and Format

WEDELMUSIC is an XML compliant format which includes constructs for the description of integrated music objects. Digital music objects compliant with the WEDELMUSIC format are called WEDELMUSIC or simply WEDEL objects. The model is supported by a full set of tools for multimedia music packaging and distribution. They are focused on a specific music piece or concept. Each WEDEL object presents sections about its: identification, classification, protection, printing, symbolic music (fonts, formatting rules, versions), image score, performance, documents, animations, lyric, audio, video, and color image. Hereafter these aspects are discussed with the rational for their inclusion.

- **Identification** section allows the identification of the music piece. Typical identification codes such as ISMN, ISBN, etc., are included. Each WEDEL object has a unique identification, ID, called WDFID, while each of its components has its own code, called WDFCID.
- **Classification** section allows the classification of the music piece according to multilingual archive mechanisms integrating both the most well known standards Z39.50 and UNIMARK fields, plus other fields. Distinct classification records may be set up for the whole object and for its components.
- **Protection** section models details about the encryption of the WEDEL object and the watermarking of music (audio files and music sheets). According to a sophisticated Digital Right Management module, each operation which can be performed on the WEDEL object can be either permitted or inhibited; more than 150 different multimedia functionalities are managed. A permission table is available to define DRM. This model allows analysis of the end-users' needs with statistic tools since each exploited functionalities can be tracked by the distributor.
- **Symbolic Music** section describes the scoring information, musical notation symbols, and their relationships. Symbolic music can include main score and parts. The symbolic description includes specific sections for classification and identification of the music score (main score and parts). Formatting rules are formalized in MILLA language like in MOODS (Bellini, Della Santa, Nesi, 2001): positioning symbols, decision about the stem length, decisions concerning where to place symbols, whether up or down the staff, the orientation of the beam and ordering to place notation symbols around the notes: slurs, accents, ornaments, etc. The user may decide to apply different rules for the automatic reformatting of the music piece. Files in other music notation formats can also be included, for example Finale, Sibelius, Score, MuseDATA, MusicXML, etc. With additional tools it is possible to convert MIDI, Finale, SCORE and Sibelius formats into WEDELMUSIC and from this to save music notation in MIDI and Finale.
- **Image of Music Sheets** section allows to integrate images of music scores into the WEDEL object without converting them into symbolic format. Therefore, in the same WEDEL object both symbolic notation and original images can be present. The same WEDEL object may contain images of the main score and the related parts, even in more versions. This makes it possible to build WEDEL objects to compare the original music score with revised and currently used symbolic versions. It is a good support for revitalizing and recovering cultural heritage.
- **Audio** section may contain none, one or more audio files in any format. Audio files can be watermarked according to the WEDEL object's parameters.
- **Performance** section describes the synchronization aspects between each audio file and the music score of the WEDEL object. The synchronization of audio files allows the contemporaneous visualization of the score and listening to the music with the selected audio and execution rate. This can be done by using symbolic music notation or simple images of music score. A performance can also be a sequence of slides with associated an audio file.
- **Documents** section may include documents in any format. Documents may be author biography, critical description of the piece, description of the music piece, etc., any document related to the music score of the WEDEL object. This section could include also a document version of the music score. It can be in PDF or PostScript as well. Documents may have special HTTP links to other elements in the WEDEL object. PDF documents included in WEDELMUSIC present a very high level of protection since WEDELMUSIC permit the full control of the number of printouts produced by PDF files.
- **Lyric** section includes multilingual lyrics associated with the music score and therefore with the WEDEL object. Several multilingual lyrics can be associated with the same symbolic part. This is performed by using the symbolic indexing of each music notation symbol.
- **Video** section may contain video files in any format. Video files in the WEDEL object turns out to be of great help for (i) showing the hands of the musician while playing music , which is very useful for didactical purposes (hands can be visualized while the music score appears on the screen); (ii) including/showing the video of a live performance and the

songs text; etc.

- **Animations,** they can be in Flash included in HTML pages, or in PPT, or video, etc.
- **Image** section may include color images in any format. For example, the portrait of the author/performer, the picture related to the music or opera, or performer, the home/city of the author/performer, a picture of the instrument, a picture of the live performance, the CD cover, etc.

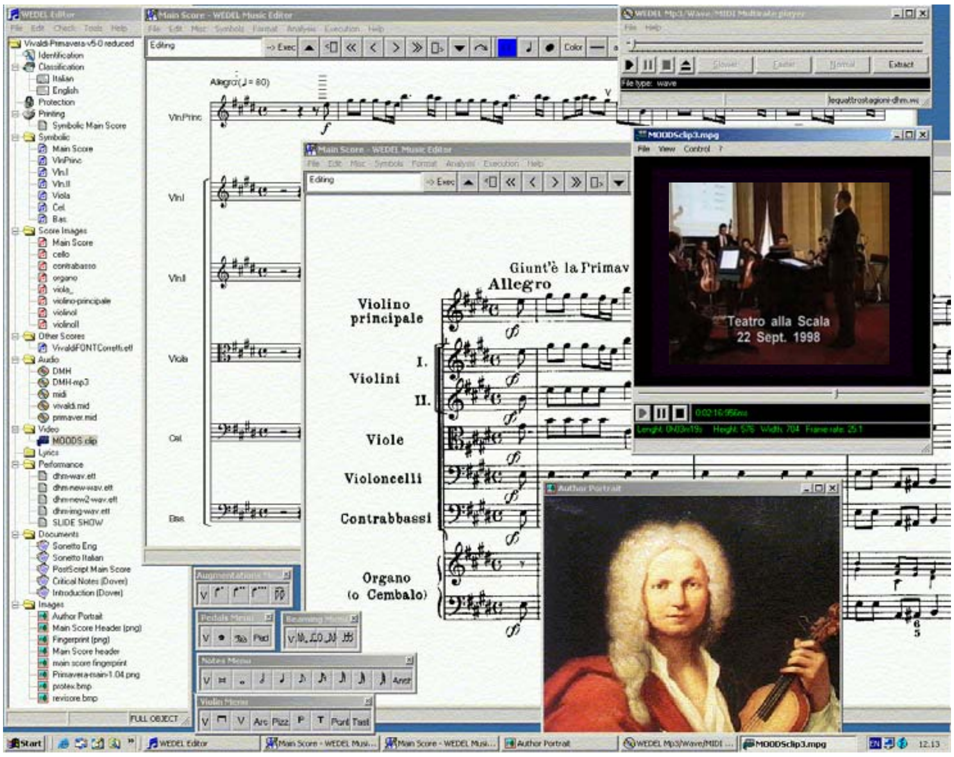

Fig.1 – An Integrated Multimedia Music Object

In each WEDEL object, several relationships among its components can be established as depicted in Fig.1.
For example, the following relationships can be established:

- Symbolic-lyric: different lyric files can be referred to the same symbolic file.
- Symbolic-image score: relationship performed thanks to the specific number associated with each measure
- Image score to symbolic: relationship performed thanks to the specific number associated with each measure.
- From Symbolic to video, images, documents, audio files: relationships performed via http links which can be assigned to music notation elements.
- Images of music score to lyric:
- From symbolic or images of music score to the audio via performance section: each audio file can be synchronized with the music score.

4.1 - Music Notation Aspects

The relationships among visual notation constructs can only be formalized considering syntax and semantic aspects. As far as music is concerned, the definition of a visual grammar is a highly complex task. To create the WEDELMUSIC formal model, we first modeled music notation components with a unified object-oriented formal model and language. In

WEDELMUSIC and MOODS, the object-oriented model has been used as a music representation model and coding language, and also as the network message interchange protocol among lecterns. In WEDELMUSIC, the logical structure of music notation is modeled in XML while the graphic details are generated in real time by using the style coded in MILLA (Bellini, Della Santa and Nesi, 2001).

MILLA can be used for defining default and exceptions rules in order to manage the already described two kinds of problems. Simple conditions can be also forced. Several symbols strictly related to notes are managed: accidentals, ornaments, fingering, accents, instrumental symbols, etc. In several instances, the mere presence of more than one of such symbol around figures changes the figures' order, orientation and position. For instance, accents must be placed on the opposite side of the stem (if any). In the event of a slur on the same note, some accents must leave space close to the note at the slur, thus they have to be placed just a bit farther from the notehead than usual (Ross, 1987), (Heussenstamm, 1987). The slur itself has to make room to other symbols when they are contemporaneously present on the same note.

In MILLA, the users can specify the following types of rules for the:

- insertion of symbols, such as the estimation of the direction (up/down with respect to the staff or with respect to the notehead) of stems, beam, slurs, etc. (they can be manually imposed if needed).
- positioning of symbols, such as the estimation of stem length, distance with respect to staff lines, beam angle, position of the ornaments, expressions, etc. with respect to the notehead.
- ordering symbols while considering the presence of neighboring symbols, ensuring precedence among symbols with respect to the notehead when depicting slurs, accents, markers, etc.;
- justification of measures according to specific algorithms, such as linear, logarithmic, and different scales; to consider alignment among voices and parts in main scores, line breaking and page breaking;
- beaming rules for beaming notes on the basis of time signature, etc.; and
- compressing symbols for the activation and deactivation of rules which display symbols in a compressed format, including generic rests and repeat symbols.

The set of MILLA rules associated with a music piece is a sort of formatting style, like the styles usually adopted in word processors. It is more sophisticated than those styles since MILLA rules can be activated according to the conditions estimated when considering local and/or general visualization contexts. Such rules typically change on the basis of the context: presence of other voices or parts, presence of other symbols, up/down the staff, etc. Most of the defined rules prevent the generation of collisions, others may permit the production of a specific collision under certain conditions (stems and beams, beams and barlines, stems and slurs, etc.). Once a new visual problem is detected, a specific rule can be added or the change can be manually performed.

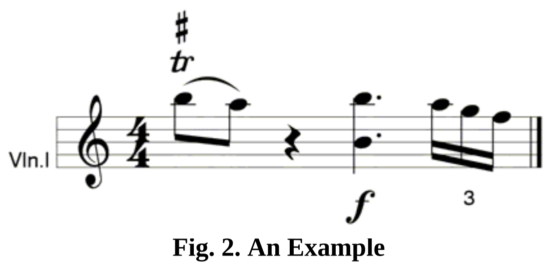

Fig. 2. An Example

In Fig.2, an example, which has been automatically formatted by WEDELMUSIC taking into account specific rules, is reported.

The slope of the two beams is determined by the following MILLA rule:

```
RULEPOS BeamInt BEAM RELNOTES ANYWHERE ANGLE=[0.15;0.3];
posIF STAFFNUMBER=1.0 AND STAFFLINES=5.0 AND DELTA<2.5 AND DELTA>0.5 THEN BeamInt;
```

The DELTA value is defined as (highest – lowest)/(number of notes -1) where highest-lowest is the distance in spaces (the distance between two staff lines) from the highest note to the lowest note of the beam.
The value of DELTA for the first beam is (11-10)/(2-1)=1 and for the second one is (10-8)/(3-2)=1, then both beams satisfy the condition of being greater than 0.5 and less than 2.5. This leads to the appliance of rule BeamInt. According to the former, the slope of the line connecting the higher and the lower note is calculated as follows: if this slope is between 0.15 and 0.3 this slope is applied; otherwise, when less than 0.15, slope 0.15 is assumed and 0.3 when greater. .
The rules used to determine the length of the stem for a chord are as follows:

```
RULEPOS Stem3_5 STEM RELNOTA LENGHT=3.5;
RULEPOS Stem4 STEM RELNOTA LENGHT=4;
RULEPOS Stem4_5 STEM RELNOTA LENGHT=4.5;
RULEPOS Stem5 STEM RELNOTA LENGHT=5.0;
RULEPOS Stem5_5 STEM RELNOTA LENGHT=5.5;
RULEPOS Stem6 STEM RELNOTA LENGHT=6.0;
RULEPOS Stem6_5 STEM RELNOTA LENGHT=6.5;
…
RULEPOS StemHeight STEM RELNOTA LENGHT=STEMHEIGHT;

posIF NOTE CROMA AND INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0 THEN Stem3_5;
posIF NOTE SEMICROMA AND INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0  THEN Stem4;
posIF NOTE BISCROMA AND INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0  THEN Stem4_5;
posIF NOTE SEMIBISCROMA AND INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0  THEN Stem5_5;
posIF NOTE FUSA AND INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0  THEN Stem6_5;
posIF INCHORD AND NUMUP>0.0 AND NUMDOWN>0.0 THEN Stem3_5;

posIF INCHORD AND NOTE STEMUP AND UPPERD>7.0 THEN StemHeight;
posIF INCHORD AND NOTE STEMDOWN AND LOWERD>7.0 THEN StemHeight;

posIF NOTE CROMA AND INCHORD THEN Stem3_5;
posIF NOTE SEMICROMA AND INCHORD THEN Stem4;
posIF NOTE BISCROMA AND INCHORD THEN Stem4_5;
posIF NOTE SEMIBISCROMA AND INCHORD THEN Stem5_5;
posIF NOTE FUSA AND INCHORD THEN Stem6_5;
posIF INCHORD THEN Stem3_5;
```

The rule applied to the chord of the example is the one evidenced, since other rules do not apply. The UPPERD and LOWERD values are the distance from the center line of the staff in spaces of the upper and the lower note of the chord. The NUMUP and the NUMDOWN values are the number of notes of the chord above and below the center line of staff.
In this case the rule applied establish that the stem has to be 3.5 spaces.

The XML encoding of the example reported in Fig. 2 is the following (some details are missing, though):


```
<?xml version="1.0" encoding="UTF-8"?>
<SWF_Part>
   <WDFCID>00000100000K</WDFCID>
   <Identification>
      <WDFID>00020000H</WDFID>
      <Publisher PUBLISHING_STATUS="OTHER">Test</Publisher>
      <Preparation_date MAJOR_VERSION="0" MINOR_VERSION="1">20021015</Preparation_date>
      <Music_Geographic_Area>Europe</Music_Geographic_Area>
   </Identification>
   <Classification xml:lang="en" Description="English">
      <Author>N/A</Author>
      <Unique_Short_Name>Example</Unique_Short_Name>
      <Title>Example</Title>
      <Genre>Classical</Genre>
      <Style>Classical</Style>
      <Original_language/>
      <Composition_date>00000000</Composition_date>
      <Epoque>
         <Start_year>0</Start_year>
         <End_year>0</End_year>
      </Epoque>
```

```
   </Classification>
   <score ID="1" TYPE="NORMAL" INSTRUMENT="Vln.I">
      <origin FROM="WEDELED"/>
      ...
      <measure PROGRESSIVE="1" ID="1">
         <justification MAINTYPE="LOG" MAINJUST="2.0" PMAINTYPE="LOG" PMAINJUST="2.0"/>
         <header>
            <clef TYPE="TREBLE"/>
            <keysignature TYPE="DOM"/>
         </header>
         <timesignature TYPE="FRACTION" NUMERATOR="4" DENOMINATOR="4"/>
         <layer NUMBER="1">
            <beam ID="10" STEMS="DOWN">
               <note ID="1" DURATION="D1_8" HEIGHT="11">
                  <ornament TYPE="TRILL" UPDOWN="UP" ACCSUP="SHARP"/>
               </note>
               <note ID="2" DURATION="D1_8" HEIGHT="10"/>
            </beam>
            <rest ID="3" DURATION="D1_4" HEIGHT="1"/>
            <chord ID="5" DURATION="D1_4" STEM="DOWN">
               <chordnote ID="5" HEIGHT="4"/>
               <chordnote ID="6" HEIGHT="11"/>
               <augmentation DOTS="1"/>
               <dynamictext DYNAMIC="F" UPDOWN="DOWN"/>
            </chord>
            <beam ID="11" STEMS="DOWN">
               <note ID="7" DURATION="D1_16" HEIGHT="10"/>
               <note ID="8" DURATION="D1_16" HEIGHT="9"/>
               <note ID="9" DURATION="D1_16" HEIGHT="8"/>
            </beam>
         </layer>
         <barline TYPE="END"/>
      </measure>
      <horizontal ID="1" TYPE="TUPLE" UPDOWN="DOWN" TUPLENUMBER="3" TUPLELINE="FALSE">
         <address MEASURE="1" LAYER="1" FIGURE="11" CHORD.OR.BEAM="7"/>
         <address MEASURE="1" LAYER="1" FIGURE="11" CHORD.OR.BEAM="9"/>
      </horizontal>
      <horizontal ID="2" TYPE="SLUR" UPDOWN="UP">
         <address MEASURE="1" LAYER="1" FIGURE="10" CHORD.OR.BEAM="1"/>
         <address MEASURE="1" LAYER="1" FIGURE="10" CHORD.OR.BEAM="2"/>
      </horizontal>
   </score>
</SWF_Part>
```

The symbolic description of a part (the score tag) is decomposed in a sequence of measures followed by the sequence of horizontal symbols (slurs, tuples, crescendo, diminuendo, etc.). Each single measure contains the header information as clef, key signature and the time signature followed by one or more layers with figures (notes, rests, beams, chords, etc.)

It should be observed that horizontal symbols refer to the starting/ending figure using a path of identifiers (measureID/layer/figureID/figureID…) for example the second note of the first beam is identified by path:1/1/10/2.

It should be observed that the HEIGHT attribute of notes/rests refers to the line/space where the note has to be positioned, 0 is the first line of the staff, 1 the first space, 2 the second line, 3 the second space etc.

WEDELMUSIC presents several innovative features that will be extremely useful for the new generation of music notation applications:

- Symbolic indexing of music notation. This makes it possible to implement selective and non-linear undo, symbolic navigation among symbols and multimedia documents (Bellini at al, 2002), management of versioning, mounting multilingual lyric.
- DRM support, including rules and watermarking support of music sheets and audio/video files. It is very important to integrate this aspect into the music model since it allow to define specific rules for versioning, music fruition, music notation printing, multimedia integration and fruition, etc.
- Integration with multimedia components such as described above. Multimedia integration can be very useful for

creating relationships among music notation symbols and multimedia content for educational purpose and content valorization.

- Definition of custom and new symbols with their formatting rules, starting from 500 symbols.
- Management of information related to formatting the music in two different cases/views. For instance, computer view for music score reading on the computer screen during tuition and edutainment and print view as a graphic user interface for producing professional music sheets. This is performed for both the main score and each single part. This makes it possible to have, in a unique format, both main score and parts with their formatting information.

5 - Music Notation Models and Languages Comparison

Tab.1 is a scheme summary for the comparison of the above-mentioned languages and models, considering the aspects discussed in previous sections. In the table, (Y) means that the answer is not completely true or satisfactory since these features are potentially available and their presence depends on the visual/graphic editor used. In most cases, the positioning of symbols is made a bit more complex by using a set of Visualization Parameters. This approach results in imposing a single rule for the positioning of the whole score, and turns out to be a coarse simple and unsuitable solution for managing automatic positioning.

	WEDEL MUSIC	**MIDI**	**SCORE**	**MusiX TEX**	**NIFF**	**SMDL**	**Music XML**	**FINALE**	**GUIDO**
Logic model	Y	N	Y	Y	Y	Y	Y	Y	Y
Classification	Y	N	N	(Y)	(Y)	Y	(Y)	(Y)	N
Identification	Y	N	N	(Y)	(Y)	Y	(Y)	(Y)	N
Visual	Y	N	Y	Y (print)	(Y)	N	Y	Y	Y
Visualization Rules	Y (Milla)	N	N	N	N	N	N	(N)	N
Visualization Parameters	Y	N	Y	N	Y	N	N	Y	(Y)
Performance or MIDI play in real time	Y	Y	N (SCORE MIDI)	N	Y	(Y)	N	Y	Y
Synchronization with real audio	Y	N	N	N	N	N	N	N	N
Images of music scores	Y	N	N	N	N	N	N	N	N
Animations	Y	N	N	N	N	N	N	N	N
Versioning support	Y	N	N	N	N	(Y)	N	N	N
Selective and un-linear undo	Y	N	N	N	N	N	N	N	N
DRM support	Y	N	N	N	N	N	N	N	N
Multimedia Integration (video, etc.)	Y	N	N	N	N	(Y)	N	N	N
Symbolic indexing of music notation	Y	N	N	N	N	N	N	N	N
Multilingual Lyric	Y	N	N	N	N	N	N	N	N
Graphic Lyric	Y	N	Y	Y	Y	Y	Y	Y	N
Stream Lyric	Y	N	N	N	N	Y	N	Y	(Y)

Tab.1 -- Coverage of Music Aspects, Y and N represent if a given feature is supported or not, when they are in round brackets means that incomplete or non satisfactory coverage is provided.

In Tab.2, the results of the comparison are reported. The information has been obtained by reviewing tools related to the languages. When a number is given, it was obtained by means of questionnaires distributed to musicians and computer scientists. In the Table, Interactive Music Editing states the availability of a visual editor; Adding/Editing Graphic Entities isthe possibility of using graphic primitives (like lines) superimposed on the score; Print support, Extraction of Parts, Extensibility of Symbols and Fusion of Parts do not need any comment; Main Score editing full length is the possibility of editing the main score as continuous information not interrupted by page breaks; Music Distribution for Cooperative work is the possibility of cooperative working during music editing or execution; Number Notation Symbols is a vote about the number and the relevance of the available notation symbols; Logic and Visual Independence is a vote about the independence of logic and visual aspects. This last vote has been obtained by analyzing the music formats and models and observing the behavior of symbols on the editors (when available) during insertion, moving and deletion.

	WEDEL MUSIC	**MIDI**	**SCORE**	**MusiX TEX**	**NIFF**	**SMDL**	**Music XML**	**FINALE**	**GUIDO**
Interactive Music Editing	Y	Y	Y	N	Y	N	(N)	Y	(Y)
Adding/Editing Graphic Entities	(N)	N	Y	Y	N	N	N	Y	Y
Print support	Y	Y	Y	Y	Y	N	N	Y	Y
Extraction of Parts	N	N	Y	N	Y	N	Y	Y	N
Fusion of Parts	Y	Y	Y	N	(Y)	N	N	(N)	N
Main Score editing full length	Y	Y	N	Y	Y	N	Y	Y	N
Automatic formatting of music, symbol positioning	Y	N	Y	Y	N	N	N	Y	(Y)
Automatic line breaking	Y	N	Y	Y	N	N	N	Y	Y
Managing Formatting information for main score and parts in the same model and file	Y	N	N	N	N	N	N	N	N
Music Distribution for Cooperative work	Y	N	N	N	N	N	N	N	N
Music Distribution tools	Y	Y	N	N	N	N	N	Y	(Y)
Music Analysis	Y	N	N	N	N	N	N	Y	N
Braille Music	Y	N	N	N	N	N	N	Y	N
Vote on the number of notation symbols: from 1 to 10 (very good)	9	4	10	8	4	4	6	9	5
Extensibility of Symbols	Y	--	Y	Y	Y	Y	N	Y	N
Logic and Visual Independence	Y	N	N	N	Y	Y	N	N	N
Code editor availability	Tool Kit	Y	N	N	N	N	N	N	Y
Liveness of editor	Monthly	stable	none	none	none	none	monthly	yearly	none

Tab.2 -- Editor, Languages and Model Comparison, Y and N represent if a given feature is supported or not, when they are in round brackets means that incomplete or non satisfactory coverage is provided

The main problems of the considered languages in meeting the requirements of the new application are mainly due to the lack of formalization in the language and in the model for storing/coding music. Concerning the definition of relationships among symbols, the most flexible and complete language is NIFF. Even this language is not fully satisfactory since it does not model all the relationships. The real problem of these languages is the lack of versioning support and the management of visualization rules. These two aspects are also missing in the several new mark-up languages which have been recently proposed on the WWW. They are presently strongly limited compared with the above models. Most of them do not present slurs, accents, instrumental symbols, justification, etc. There are also several other types of languages for coding music, as described in Selfridge-Field (1997), but unfortunately, none of them can be held as a complete support for the requirements of the new incoming applications.

On the account of the work performed on several projects and products, some tools for operating the conversion among these formats have been produced. Tab.3 summarizes the available converters among the considered languages. Some of them exist only at a hypothetical level because they are for instance based on claimed prototypes but the converter is not distributed or the conversion has been only studied and analyzed. In all these conversions, several details are lost since different formats treat the information in a different manner, others present a limited number of symbols and relationships, thus being forced to eliminate information during the transformation.

From\ to	**WEDEL MUSIC**	**MIDI**	**SCORE**	**MusiX Tex**	**NIFF**	**SMDL**	**Music XML**	**FINALE**	**GUIDO**
WEDELMUSIC	--	Y	N	N	N	N	N	Y	N
MIDI	Y	--	Y (MIDISCORE)	N	Y (Lime)	Y	Y	Y	Y
SCORE	Y via Sibelius	Y	--	N	N	N	N	N	N
MusiXTEX	N	N	N	--	N	N	N	N	N
NIFF	N	Y (Lime)	N	N	--	(N) (Cantate)	Y	N	N
SMDL	N	N	N	N	(N) (Cantate)	---	N	N	N
MusicXML	N	Y	N	N	N	N	…	Y	N
FINALE	Y	Y	Y (Final SCORE)	N	N	N	Y	…	(Y) only few symbols
GUIDO	N	Y	N	N	N	N	N	N	….

Tab.3 -- Available Format Converters, Y and N represent if a given feature is supported or not, when they are in round brackets means that incomplete or non satisfactory coverage is provided

From the above comparison it evident that WEDELMUSIC is the best candidate for managing innovative and new emerging multimedia applications. It contains several innovative aspects at level of music notation such as symbolic indexing, strong distinction between logic and graphics, MILLA formatting engine, multilingual lyric, etc.

5.2 -- Future Trends of Music Notation

Music notation and music notation programs are at present passing through the same evolution that text and text editors went through several years ago. Music publishers are still focused on obtaining good quality (in terms of visual formatting) of music scores. On the other hand, the market would like to have lower prices and considers the quality not so important. Publishers state that they sell only high quality music scores, which is quite obvious when prices are so high. Musicians are used to reading very poor music scores in terms of both formatting and visual resolution. To grasp this, it is enough to observe the quality of music sheets used by musicians in the conservatories and music schools. Their quality is really poor, they are frequently photocopies of photocopies of old originals. Sometimes it can even occur that the music is unreadable in some very ancient pieces.

The market is ready for massive production of music sheets while publishers are not ready for it. The profit margins are in the high quality since the numbers are small. At the same time students and music lovers prefer to make photocopies. This is the same story as it happens with books. Publishers have recovered their market only by producing cheap books with low quality paper and formatting shapes.

For these reasons, the future of this field is in the automatic formatting of music sheets and in the delivering of this via Internet. Huge problems have to be solved for modeling music and for integrating this with the multimedia aspects of music. In Europe a special project and service has been set up by the European Commission to stimulate this integration. This project is the MUSICNETWORK, www.interactivemusicnetwork.org.

6 - Conclusions

The WEDELMUSIC model and language has been defined after an analysis of several other coding approaches to look for a model to cover the needs of some new emerging applications like the cooperative editing of music in theatres/music schools and the distribution of music via Internet, versioning, selective and non linear undo, educational tool. The WEDELMUSIC presents a clear distinction between logic and visual parts. The latter is defined by means of MILLA rules. An early version of the WEDELMUSIC format was MOODS format. In its early version it was provided without MILLA and that experience has convinced us to work on a separate and independent engine for music formatting, with the aim of: (i) providing different rules that can be applied in different visual conditions along the same score, (ii) making possible the definition of formatting styles.

We do not claim to have solved all the problems related with music automatic formatting. On the other hand, MILLA can be used for defining rules for the automatic formatting of music scores with similar problems and conflicts. WEDELMUSIC editor is freely distributed on the www site: www.wedelmusic.org. The WEDELMUSIC development kit (WTK) is also freely distributed for loading and saving music in WEDEL format including audio, image and symbolic aspects, integrated together. An annual International Conference on WEB Delivery of Music is being organized. WEDELMUSIC format is royalty free, the format is public and available in DTD in the www site of WEDELMUSIC.org.

Acknowledgments

The authors thank all partners, members, experts and participants of the MOODS, the WEDELMUSIC (www.wedelmusic.org), and the MUSICNETWORK (www.interactivemusicnetwork.org) projects, a special thank to Gerd Castan for his valuable comments that have permitted to better explain the comparison model and its motivation.. A sincere thanks to the constructive comments provided by the reviewers of this chapter.

References

Anderson, D. P. and Kuivila, R. (1991). Formula: A Programming Language for Expressive Computer Music. IEEE Computer, pp.12-21, July.

Bellini, P, Della Santa, R., Nesi, P. (2001). Automatic Formatting of Music Sheet. Proc. of the First International Conference on WEB Delivering of Music, WEDELMUSIC-2001, IEEE Press, 23-24 November, Florence, Italy, pages 170-177.

Bellini, P., Bethelemy, J., Bruno, I., Nesi, P., Spinu, M. B. (2003). Multimedia Music Sharing among Mediateques, Archives and Distribution to their attendees. Journal on Applied Artificial Intelligence, Taylor and Francis, in press, www.wedelmusic.org.

Bellini, P., Fioravanti, F., Nesi, P., (1999). Managing Music in Orchestras, IEEE Computer, pp.26-34, September, http://www.dsi.unifi.it/~moods/.

Bellini, P., Nesi, P. (2001). WEDELMUSIC FORMAT: An XML Music Notation Format for Emerging Applications. Proceedings of the 1st International Conference of Web Delivering of Music. 23-24 November, Florence, Italy, pages.79-86, IEEE press.

Bellini, P., Nesi, P., Spinu, M. B. (2002). Cooperative Visual Manipulation of Music Notation. ACM Transactions on Computer-Human Interaction, September, 9(3):194-237,

http://www.dsi.unifi.it/~moods/.
Blostein, D. and Baird, H. S. (1992). A Critical Survey of Music Image Analysis. Structured Document Image Analysis, (H. S. Baird and H. Bunke and K. Yamamoto, ed.), Springer Verlag, NewYork, USA, pp.405-434.
Blostein, D. and Haken, L. (1991). Justification of Printed Music. Communications of the ACM, 34(3):88-99, March.
Byrd, D. A., (1984). Music Notation by Computer. Department of Computer Science, Indiana University, USA, UMI, Dissertation Service, http://www.umi.com.
CANTATE project, (1994). Deliverable 3.3: Report on SMDL evaluation, WP3.
Dannenberg, R. B. (1990). A Structure for Efficient Update, Incremental Redisplay and Undo in Graphical Editors. Software Practice and Experience, 20(2):109-132, February.
Dannenberg, R. B. (1993). A Brief Survey of Music Representation Issues, Techniques, and Systems. Computer Music Journal, 17(3):20-30.
Good, M. (2001). MusicXML for Notation and Analysis. In W. B. Hewlett & E. Selfridge-Field (Eds.), The Virtual Score Representation, Retrieval, Restoration (pp.113-124). Cambridge,MT: The MIT Press.
Gourlay, J. S. (1986). A Language for Music Printing. Communications of the ACM, 29(5):388-401, May.
Heussenstamm, G. (1987). The Norton Manual of Music Notation, Norton \& Company, New York, London.
Icking, I. (1997). MuTEX, MusicTEX, and MusiXTEX Beyond MIDI - The Handbook of Musical Codes, (E. Selfridge-Field, ed.), The MIT Press, London, pages 222-231.
NIFF Consortium, (1995). NIFF 6a: Notation Interchange File Format.
Pennycook, B. W. (1985). Computer-Music Interfaces: A Survey, ACM Computing Surveys, 17(2):267-289, June.
Pope, S. T., (1991). The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology. MIT press.
Rader, G. M. (1996). Creating Printed Music Automatically, IEEE Computer, pp.61-68, June.
Ross, T. (1987). Teach Yourself. The Art of Music Engraving. Miami, London: Hansen Books.
Selfridge-Field, E. (Ed.). (1997). Beyond MIDI - The Handbook of Musical Codes. London, UK: The MIT Press.
Sloan, D. (1997). HyTime and Standard Music Description Language: A Document-Description Approach. E. Selfridge-Field (Ed.), Beyond MIDI - The Handbook of Musical Codes. (pages 469-490). London, UK: The MIT Press.
SMDL ISO/IEC. (1995). Standard Music Description Language. ISO/IEC DIS 10743.
Smith, L. (1997). SCORE. Beyond MIDI - The Handbook of Musical Codes, (E. Selfridge-Field, ed.), The MIT Press, London, pages 252-282.
Taube, R. (1998). CCRMA, Common Music.' CCRMA, Stanford University, California, USA.
Taupin, D., Mitchell, R., Egler, A. (1997). Using TEX to Write Polyphonic or Instrumental Music ver T.77,'hprib.lps.u-psud.fr.
Wood, D. (1989). Hemidemisemiquavers...and other such things. A concise guide to music notation, The Heritag Music Press, Dayton, Ohio, USA.

Biographies

Pierfrancesco Bellini *is a contract Professor at the University of Florence, Department of Systems and Informatics. His research interests include object-oriented technology, real-time systems, formal languages, computer music. Bellini received a PhD in electronic and informatics engineering from the University of Florence, and worked on MOODS, WEDELMUSIC, IMUTUS, MUSICNETWORK projects of the European Commission.*

Paolo Nesi *is a full professor at the University of Florence, Department of Systems and Informatics. His research interests include object-oriented technology, real-time systems, quality, system assessment, testing, formal languages, physical models, computer music, and parallel architectures. Nesi received a PhD in electronic and informatics engineering from the University of Padoa. He has been the general Chair of WEDELMUSIC conference and of several other internal conferences. He is the coordinator of the following Research and Development multipartner projects: MOODS (Music Object Oriented Distributed System, http://www.dsi.unifi.it/~moods/), WEDELMUSIC (WEB Delivering of Music Score, www.wedelmusic.org), and MUSICNETWORK (The Interactive Music Network, www.interactivemusicnetwork.org). Contact Nesi at nesi@dsi.unifi.it, or at nesi@ingfi1.ing.unifi.it.*