Perry Roland
Digital Library Research Group
University of Virginia

Abstract: This paper evaluates the role of standards in information exchange and suggests the adoption of XML standards for music representation and meta-data to serve as the basis for music information retrieval.

Keywords: Music Information Retrieval (MIR), eXtensible Markup Language (XML), music representation, information exchange standards, Unicode, MusiCat

XML4MIR:
Extensible Markup Language for Music Information Retrieval

The most important word in the phrase “Music Information Retrieval” is “information”. Ponder why it’s not “Music Data Retrieval”. Information is not just data. Instead, information is communication, the exchanging of data and meta-data. A great deal of effort has been expended on capturing music data; however, for the full potential of MIR to be realized, we must concentrate on how the communication of the data will take place. Standards, e.g. common languages, are the enablers of communication.

Why do we need standards?

Standards are good for scholars. They extend the scale, breadth, and accessibility of scholarly evidence and encourage innovation in learning and teaching. In addition, standards facilitate innovation and collaboration in scholarly discourse (Greenstein, 254).

Standards are also good for business. They reduce the costs associated with acquiring and preparing data and help create new sources of revenue for that data. For example, publishers can syndicate material to other information providers or target their material to traditional publishing formats or even electronic formats, such as Internet devices or CD-ROMs, more easily. Additional revenue can be generated from the same material with minimal cost and effort (Floyd, 48).

Bob Metcalfe has stated "the usefulness of a network grows exponentially with its number of users" (qtd. in St. Laurent). The clearest example of this statement in action is the rapid growth of the Internet. Of course, Metcalfe's Law can be applied to other technologies as well, including data encoding standards: With a standard, more users will take advantage of the technology; as the number of users increases, the price of implementing the technology falls; falling prices create competition to improve the technology; and, finally, improved technology increases the usefulness of the standard.

The implications of this economic model of technology are that continuing the status quo, without a standard, is shortsighted. In the long-term, more readily available data means more customers, at least for tools and value-added information, if not for the raw data itself. At least one company, Muze Incorporated, which plans to provide electronic meta-data for music (Luh), believes that more efficient access to information may also speed up consumers' purchase

decisions, presumably creating more purchases. Also, in this economic model, both commercial (publishing) and non-commercial (scholarly) users of the technology benefit as they each function in a symbiotic relationship as both data providers and data consumers.

Why not use existing standards?

While it might be possible to adopt an existing music representation, there are several problems with this approach.

Most existing music representations are inappropriate due to their scope. The analytical domain, which we seek to exploit in MIR, is most often the first thing to be defined as "out of scope". Many representations define their approach to music encoding too narrowly, concentrating on a particular use of the data such as printing or automated performance. Most of these representations are useful as input codes, but they have limited use as intermediate representations (Selfridge-Field, 571). Other representations, such as the Standardized Music Description Language, have attempted to represent music too broadly. SMDL has been unable to attract a large user group in part because it is difficult for potential users and tool developers to see how SMDL might apply to their particular situation. A standard that is defined generally enough to represent all music can only be made to work for a particular subset with great effort.

In addition to problems of scope, many existing solutions are hardware or software dependent. The lack of acceptance of specialized hardware input devices for music, such as tablets and touch screens, outdated storage mechanisms, like punched cards and paper tape, and incompatible file formats should stress the necessity of hardware neutrality.

Many existing codes are also proprietary. After expending a great deal of effort to create them, their owners are reluctant to divulge their inner workings. Therefore, their use for information exchange is severely limited.

Not only is there a need for a music representation standard; we also need to examine existing meta-data standards in order to judge their suitability for MIR purposes.

As compromise solutions for a wide variety of materials, most meta-data systems are inadequate for MIR. For example, using MARC (MAchine Readable Catalog) records it is difficult to express the complex relationships frequently found in music meta-data. The TEI (Text Encoding Initiative) scheme is designed for text representation, not meta-data exchange. The EAD (Encoded Archival Description) format is designed for the use of the archival community and so lacks elements necessary for the complete description of music objects. For these reasons, a meta-data standard for music has been proposed which not only addresses these issues, but also allows the leveraging of existing music meta-data, e.g. thematic catalogs (Roland, MusiCat).

Why use XML?

The foremost reason for using XML is that it is a platform-independent, open standard. XML is a simplified subset of the Standard Generalized Markup Language (ISO8879) standard. Despite its name, XML isn't really a markup language, but a meta-language, designed to support the definition of community-specific languages. Because there are no limits on the use of elements across multiple namespaces or on the structural depth that a markup language might employ, XML is very powerful. Furthermore, it is easy to implement. Developers need not create their own tools for authoring, parsing, transforming or displaying XML. There is already an ever-growing set of free tools available.

The fact that the SGML/XML approach is nearing ubiquity also provides a strong reason for employing it for music representation and meta-data. XML is the foundation of a broad range of industry and academic standards, such as CML (chemistry), MML (mathematics), and ThML (theology), PGML and SVG (graphics), and SMIL (presentation media). Although most of them are not yet fully developed, there are even a few XML markup schemes for music.

Outside the music field there is a large, organized SGML/XML community already in existence. Users include academic institutions, government agencies, such as the U.S. Government Printing Office, Department of Defense and Internal Revenue Service, as well as commercial interests such as publishers, airlines, and manufacturers. Organizations promoting the development of XML include ISO (International Standards Organization), OASIS (Organization for the Advancement of Structured Information Standards), and W3C (World-Wide Web Consortium). A large community of SGML/XML users indicates a potential consumer base for encoded music materials. It also means that the resources and skills of this community are transferable to the task of encoding music.

Internationalization provides another argument in favor of adopting XML. XML was designed from the beginning to support the inclusion of multiple languages and symbols. Since music often includes text as labels, lyrics and performance directions, it is appropriate to employ an encoding that uses Unicode, an extensive international character set, for both content and markup.

Why use XML for music representation?

XML is grammar-based. XML documents can be validated using a Document Type Definition or DTD, which is a formal statement of the rules governing the document's grammar. Roads enumerates several advantages of grammars: there is a long history of research into the use of grammars in music description, because any music that can be segmented can be represented, grammars are very powerful, grammars provide a generative model (Roads, 408-409). In addition, utilizing a grammar speeds encoding and reduces encoding errors.

XML is declarative. According to Desain and Honing, a declarative representation is preferable to a procedural one because declarative knowledge is accessible. That is, it can easily be examined and combined with information from other sources. A declarative representation is also composable, e.g. the meaning of a complex expression is based on or can be derived from the meaning of its parts and their combinations (Desain and Honing, 142). Also, since no interactions occur between structural entities, the representation is extremely modular (Desain and Honing, 131). The importance of modularity is discussed in more detail below.

XML is structured hierarchically. The hierarchical view of music suggested by Buxton seems natural to most musicians because they are trained to approach music this way. Teachers often implore students to perform phrases, not notes. Similarly, when we analyze the structure of a piece of music, it is natural to name its parts and show how some elements are more important than others. This is the essence of analysis. In his design principles for music representation, David Huron (Huron1992, 26) has suggested that it is appropriate for representation to be isomorphic with the thing represented. One might say XML is structurally isomorphic with music.

In addition, the tree data structure is conceptually easy and provides efficient, non-linear data retrieval. Also, this data structure makes it possible to apply transformations to groups of objects (Gourlay, 393).

Of course, the simplest hierarchy, e.g. a sequence of events, is also conceptually simple, but it is difficult to encode structural relationships between events using a single sequence. Defining the structural relationships in music is perhaps the most important goal of representation. Music encoded as a sequence of events (e.g. as in MIDI) makes automated error detection impossible when more than one kind of event is included.

On the other hand, the most problematic thing about defining structure in music is that a single hierarchy is inadequate to describe the multiple, and sometimes overlapping, hierarchies such as those created by beams, measures, phrases, voices, and sections. However, the SGML/XML community has developed several methods of representing multiple hierarchies in texts (Barnard, et al) that can be applied to music. First, attributes (or perhaps other elements) can be used to mark an element as belonging to one or more hierarchies. Figure 1 shows how an elision of two phrases might be indicated using attributes.

```
<note id="n1" hier="phrase1"/>
<note id="n2" hier="phrase1"/>
<note id="n3" hier="phrase1 phrase2"/>
<note id="n4" hier="phrase2"/>
<note id="n5" hier="phrase2"/>
<note id="n6" hier="phrase2"/>
```

Figure 1

Second, boundary elements that have no content themselves can be used to mark the beginning and end of events. Figure 2 illustrates the use of these elements to mark the same elided phrases. Note that the position of the phrase-defining elements is significant.

```
<beginphrase id="p1"/><note id="n1"/><note id="n2"/>
<beginphrase id="p2"/><note id="n3"/><endphrase id="p1"/>
<note id="n4"/><note id="n5"/><note id="n6"/><endphrase id="p2"/>
```

Figure 2

Finally, separate analysis elements can be used to create references that show how other elements are to be organized. In figure 3 the position of the phrase elements is irrelevant to the organization of the notes. They could be placed elsewhere in the document or, using other XML-related standards, such as XML Linking Language (XLink) and XML Path Language (XPath), even in a separate document.

```
<note id="n1"><note id="n2"><note id="n3">
<phrase id="p1" start="n1" end="n3"/>
<note id="n4"><note id="n5"><note id="n6">
<phrase id="p2" start="n3" end="n6"/>
```

Figure 3

Of course, more than one of these approaches could be implemented at a time, giving the user the opportunity to select the method most appropriate for the music being encoded. Also, even though it introduces a certain amount of inconsistency, different methods can be applied to different hierarchies. For instance, phrases might be marked with analysis elements while beams are indicated by boundary elements.

XML is modular. Music is often thought of as having separate visual, analytical, and performance aspects or domains. Furthermore, each of these domains is frequently described in

terms of separate facets, e.g. time, pitch, harmony, etc. Of course, not every musical task involves all the domains or facets. Printing perhaps requires the most representationally complete encoding, while analysis requires less, and automated performance less still. Using XML would facilitate a general encoding which allowed the facets most important to the task at hand to be represented completely and those of less importance to be minimized or left out entirely, a concept Huron calls "selective feature encoding." (Huron1997, 375) In other words, the XML Document Type Definition can be written to allow multiple levels of representational completeness. A modular DTD also allows multiple encoding styles. For example, for representing pitch one encoder prefers MIDI note numbers while another would rather use ANSI standard pitch designations such as "C#4". Either style can be allowed in the representation simply by "switching on" the appropriate module. Of course, each of the other domains and facets can be handled in a similar fashion.

XML is extensible, an absolute requirement for music representation systems. Extensibility functions to "resolve ambiguity already latent within the existing scheme" (Huron1992, 35). Extensibility also allows the representation to absorb some changes in coding requirements. For example, a DTD can provide a mechanism for making arbitrary changes to element names and content specifications. This feature might be useful for translating English element names into Russian for a colleague in Moscow or redefining the content model for note elements to accommodate future developments, such as a successor to MIDI.

As long as the extensions are simply additions, the representation may be extended without necessarily making existing documents non-conformant. Of course, deciding when to extend and when to abandon the representation for a new one becomes difficult when the extensions affect the structure of the representation. The ultimate in extensibility, however, is a good exit strategy. Since a DTD is not absolutely required for XML, one could opt for a well-formed representation, that is, where there are no overlapping markup structures or, with the tools discussed below, transform the old grammar into a new one.

XML is human-readable. Human-readability makes data creation and maintenance easier and functions as a protection against technological obsolescence. These considerations are very important for music because the body of material to be encoded is so vast and the investment in encoding is so large.

At least two of Huron's design principles are addressed by human-readable encoding. First, an encoding that is human-readable can be made mnemonic, that is, meaningful to the encoder who must learn little or no new vocabulary. Second, human-readable representations are non-cryptic. They make it easier to use the encoding because the relationship between the representation and the thing represented is made clear. They also contain redundancy that aids in the recognition of errors (Huron1992, 26).

```
IML
'=CLEF G' '*KEY UFS LC' '+TIME(C)'
LD2 UE4 *F4 / F4* LD8 C8 UUA4.  G8 / FN1 //

PLAINE AND EASIE CODE
(#FC, C) '2D 4E F_/ 8D C ''4.A 8G / 1NF //

DARMS
!G !K2# !MC 20H 1Q 2J / 2 (20 19) 31Q 30E/ 9*W /!/
```

```
XML CODE
<mdl>
      <clef type=”G” pos=”treble”/>
      <key sig=”2#” tonic=”D”/>
      <time sig=”C” norm=”4/4”/>
      <bar startline=”invis”>
            <note pitch=”D4” dur=”2”/>
            <note pitch=”E4” dur=”4”/>
            <note pitch=”F#4” dur=”4” tie=”initial”/>
      </bar>
      <bar>
            <note pitch=”F#4” dur=”4” tie=”final”/>
            <note pitch=”D4” dur=”8”/>
            <note pitch=”C#4” dur=”8”/>
            <note pitch=”A5” dur=”4.”/>
            <note pitch=”G5” dur=”8”/>
      </bar>
      <bar endline=”endbar”>
            <note pitch=”Fn5” dur=”1”/>
      </bar>
</mdl>
```

Figure 4

Figure 4 illustrates the impact of human readability. While it cannot be said that the XML code is more compact than the others, it is clearly not a private code. Even to someone with minimal training, it is immediately obvious what the code represents. Furthermore, unambiguous labeling of the data also enhances machine readability.

While XML may be somewhat verbose, reduction of the code can be achieved by several methods. User-defined entities and standard character entity references may be employed. A list of 220 music symbols (Roland) has been approved by the Unicode Technical Committee and is awaiting the next step on the path to approval by the International Standards Organization. Code reduction for network transmission can be accomplished via standard loss-less compression schemes, such as Run Length Encoding (RLE) and Lempel-Ziv-Walsh (LZW) compression, and by network acceleration technology currently in development (XML Growth). However, as processors get faster and more bandwidth becomes available, terseness for machine processing and transmission will become less of a concern. We must be careful not to repeat the mistakes of the past, i.e. encoding of dates with two digits instead of four, when immediate processing needs were allowed to take precedence over protecting the data from technological changes.

The final, and perhaps most important, reason for choosing XML for MIR is that it separates content and structure from presentation and behavior. Taking data creation into the equation, the problem of music representation can be seen as consisting of three components -- input, communication, and output (Gourlay, 389). Separation into these components alleviates the problems of strong coupling of the syntax and semantics of the music encoding language to particular processors or processing techniques (Page, 48-49). The main disadvantage of decoupling is that additional mechanisms are required for associating behavior with the data. Industry agreement on the creation and processing of a specific grammar is required -- no small task given that music notation is complex, ambiguous, and redundant, and that there is a very wide range of processing requirements. Fortunately, companion standards to XML, XSLT and XPath, offer standard methods for processing, transforming, and querying XML documents.

The advantages of decoupling content from presentation outweigh the disadvantages. First, high-level abstraction of the content allows particular representations to be generated as needed. The content can be re-used more effectively since the creator provides a single data stream for both data exchange, e.g. machine consumption, and publishing, e.g. human consumption. Media independence in publishing can be achieved without altering the basic data. Different devices, such as web browsers and MIDI file players, may render the same data differently. Downstream document processing, such as that required for some MIR tasks like displaying selected portions of the data, may be carried out independently of the data. User autonomy, an important consideration in music representation (Dannenberg, 24; Byrd, 20) can be provided through user-configurable views of the data. Second, rendition on the client reduces the server load and transmission time in a network environment. Finally, accommodating a large number of encoders is essential for encoding the massive amount of music required for large-scale MIR. By separating the input and communication phases, input can be accomplished using tools, such as word processors, databases, MIDI devices, graphical interfaces, or text processors, which accommodate a wide range of user preferences and budgets (Gourlay, 391).

Why use XML for music meta-data?

All of the advantages of XML enumerated above apply not only to music representation, but also to the representation of music meta-data. The primary advantage of adopting XML for both is improved integration of the music and its meta-data. XML music representations may contain XML meta-data or XML meta-data may contain XML music representations. Figure 5 illustrates how common elements can be shared between the two structures. In the MusiCat markup the notation is used to uniquely identify the object of the meta-data while in the mdl markup the notation is a complete representation of the principal object.

```
<MusiCat>
      <title>Musicke in Bbb</title>
      <agent>Roland, Perry</agent>
      <incipit>
            <notation>(...)</notation>
      </incipit>
      <analysis>(...)</analysis>
</MusiCat>
```

```
<mdl>
      <title>Musicke in Bbb</title>
      <agent>Roland, Perry</agent>
      <notation>
            <note/>
            <note/>(...)
      </notation>
</mdl>
```

Figure 5

To summarize, XML provides the music community with a method for achieving interoperability of content and style, freedom from vendor control of the data, creator control of the markup syntax, and user control of the behavior of the data. Given the complexity of the task of creating a large-scale music information retrieval system, these are advantages that we cannot afford to disregard.

Works Cited

"XML Growth Continues -- In More Ways Than One." IT Director 10 May 2000. Available: http://www.it-director.com/00-05-10-2.html. May 12, 2000.

Barnard, David T., et al. "Hierarchical Encoding of Text:Technical Problems and SGML Solutions." *The Text Encoding Initiative: Background and Contents*. Guest Ed. Nancy Ide and Jean Véronis. Computers and the Humanities 29 (1995): 211-231. Available: http://www.oasis-open.org/cover/barnardHier-ps.gz. May 12, 2000.

Buxton, William, et al. "The Use of Hierarchy and Instance in a Data Structure for Computer Music." Foundations of Computer Music. Ed. Curtis Roads and John Strawn. Cambridge, MA: MIT Press, 1985. 443-466.

Byrd, Donald. "Music Notation Software and Intelligence." Computer Music Journal 18 (Spring 1994): 17-20.

Dannenberg, Roger B. "Music Representation Issues, Techniques, and Systems." Computer Music Journal 17 (Fall 1993): 20-30.

Desain, Peter and Henkjan Honing. Music, Mind and Machine: Studies in Computer Music, Music Cognition and Artificial Intelligence. Amsterdam: Thesis Publishers, 1992.

Floyd, Michael. "Separating Body from Soul." Web Techniques 5 (July 2000): 46-49.

Gourlay, John S. "A Language for Music Printing." Communications of the ACM 5 (May 1986): 388-401.

Greenstein, Daniel. "Publishing Scholarly Information in a Digital Millennium." Computers in the Humanities 32 (1998): 253-256.

Huron, David. "Design Principles in Computer-Based Representation." Computer Representations and Models in Music. Ed. Alan Marsden and Anthony Pople. New York: Academic Press, 1992. 5-39.

Huron, David. "Humdrum and Kern: Selective Feature Encoding." Beyond MIDI: The Handbook of Musical Codes. Ed. Eleanor Selfridge-Field. Cambridge, MA: MIT Press, 1997. 375-401.

Luh, James C. "A Company Trying to Drive a Real-World Use of XML." Internet World 18 Jan. 1999: 17.

Page, Stephen Dowland. Computer Tools for Music Information Retrieval. Diss. Oxford Univ., 1988. Ann Arbor: UMI, 1988. AAGD-96200.

Roads, Curtis. "Grammars as Representations for Music." Foundations of Computer Music. Ed. Curtis Roads and John Strawn. Cambridge, MA: MIT Press, 1985. 403-442.

Roland, Perry. MusiCat DTD. Latest version available: http://www.lib.virginia.edu/~pdr4h/MusiCat/DTD/.

Roland, Perry. "Proposed Musical Characters in Unicode (ISO/IEC 10646)." Beyond MIDI: The Handbook of Musical Codes. Ed. Eleanor Selfridge-Field. Cambridge, MA: MIT Press, 1997. 553-562. Latest version available: http://www.lib.virginia.edu/dmmc/Music/UnicodeMusic.

Selfridge-Field, Eleanor. "Beyond Codes: Issues in Musical Representation." Beyond MIDI: The Handbook of Musical Codes. Ed. Eleanor Selfridge-Field. Cambridge, MA: MIT Press, 1997. 565-572.

St. Laurent, Simon. Letting Go: The Futures of XML and SGML. Available: http://www.simonstl.com/articles/lettinggo.htm. May 12, 2000.

Unicode Consortium. The Unicode Standard, Version 3.0. Reading, MA: Addison-Wesley, 2000.