A Music-Workstation Based on Multiple Hierarchical Views of Music

Mori Balaian
Department of Mathematics and Computer Science
Ben-Gurion University
P.O. Box 653, Beer Sheva 84105, ISRAEL
mori@benga.bitnet

ABSTRACT.
The input/output channel is a long time problem of theoretical computer systems in music research. Musicians aid tools, on the other hand, provide convenient input/output channels, but are based on weak and task specialized data structures.

This paper describes the initial steps in building a music workstation, that is based on a view of music as a (possibly multiple) hierarchical structure, organized over a time scale. This view captures the basic notion of grouping, found in any aspect of music, while avoiding further assumptions about level, kind, or form. Therefore it seems appropriate for standardizing music processing for both, theoretical and practical purposes.

The temporal-hierarchical view is realized by a data structure called music structure, and is further extended by an attribute mechanism that provides information about the structured objects. Attributed music structures subsume all music representations known to the author. The music workstation is currently under development in COMMON LISP on a SUN 3/20 machine.

1. Background

A central issue in every simulation of a real world environment is: how to represent that environment? The power, or lack thereof, that result from an appropriate, or inappropriate, data-structure or representation formalism is well recognized in computer science in general, and in AI, in particular.

In the field of computer music, representational issues play a major role in systems geared towards analytical or cognitive purposes ([Laske 1972], [Laske 1975], [Wiener 1961], [Frankel et al. 1978], [Smolar 1976], [Smolar 1980], [Eco 1986], [Kassler 1977], [Lidov & Gabora 1973], [Lewin 1986], [Minsky 1981], [Rahban 1986]). The traditional problem of these systems is their input/output (I/O) bottleneck. Musicians aid tools (compositional tools, music-workstations, tutoring systems, typesetting programs), on the other hand, usually provide convenient I/O channels, but, are based on weak, task specialized, and typically poorly formulated data structures. These systems cannot be combined together so to extend their scopes, and if standardization is at all possible, it would be on the stream-like, structureless level of
In commercial systems, like those discussed in [Yarivov 1986], Personnel-compose ([Milo 1985]), Mockingbird ([Maxwell & Oranien 1983]), Muspritz ([Hamel 1986]), to name just a few, the representation of the user environment, i.e., the music, is deeply embedded in the system. In more research oriented systems like DARMS ([Erickson 1975]), SMUT ([Byrd 1977]), MIDI ([Loy 1985]), [Courlay 1986], PLÀ ([Schottstaedt 1983]), FORMS ([Robert & Cote 1984]), SSSY ([Buxton 1978], [Buxton et al. 1978]), IOS ([Roads 1983]), ARCTIC ([Dannenberg et al. 1986] and [Hamel et al. 1987]), straightforward off-the-shelf data structures are used to represent the music. The music material is, typically, viewed as a possibly multi-dimensional, sequence of notes, augmented with print or performance information about instruments, key signatures, and the like. Explicit structuring of the music, and inclusion of abstract musical concepts is not common. Exceptions are [Buxton et al. 1978] that uses hierarchical structures, though they are embedded in the program; FORMS where the music structure is realized by the structure of the objects generated through a program run, and ARCTIC that uses functional representation.

This lack of standardization leads to, what Courlay, in [1986], calls, "the communication problem": Users of different music processors are unable to communicate and share files; the music composed by a composition system cannot be fed into analysis programs; music analysis and music instruction cannot cooperate, etc.

The problem is that being developed with a specific task in mind, these systems do not lend themselves for generalization, either because their data structures are embedded in programs or because they are too task oriented. A program designed to achieve a specific goal, naturally, concentrates on the relevant aspects only; its data structures, even if well formulated, are specialized for the music window under consideration. Yet music is a multiple aspects phenomenon, and in order to standardize its processing there is a need for a representation that is flexible enough to account, simultaneously, for various views of a piece.

In this paper, we introduce a view of music pieces as (possibly multiple) hierarchically structured, organized along a time scale, and provide a representation, called music structures, that can capture our approach. This representation is iterated by an attribute mechanism, that can characterize the structured objects, and provide information about them. We claim that music structures are appropriate as a standard for music representation, since they concentrate on the essential (arbitrary) hierarchical structure inherent to almost any viewpoint of music; there is no commitment to a particular structure or a taxonomy of structures; no particular theory, kind, or style of music is assumed: no particular musical task is involved. A music piece can be described, simultaneously, from many viewpoints: from typsetting to theoretical analysis. Due to these features, music structures are expressive, flexible to expansions, and conceptually appealing. In fact, music structures subsume all the music representations used in the mentioned above systems.

Historically, music structures emerged from the Twelve Tone String defined within a more general research concerning the development of a knowledge base for the common terminology of WTM ([Balaban 1986], [Balaban 1988], [Balaban 1979]). Later on they were extended within the music workstation project, and within an AI project.

Peter Bloore and Det Rosenkrantz participated in the early stage of this project.
research concerning a calculus for describing temporal knowledge ([Balaban & Murray 1987a], [Balaban & Murray 1987b]). The version described here is, actually, a special case of the time structures, developed in the later research.

2. Music as a Temporal-Hierarchical Phenomenon

Human beings can communicate even if involved in different musical activities. The composer can transfer the results of his/her work to the publisher, the music theorist can communicate with the editor of his/her work. The music instructor can communicate musical ideas to students, etc. Behind all these communication lines, there is a common music notation, usually augmented with informal conventions about grouping the musical material in various ways. The notion of grouping discussed by music theorists reflects, probably, the realization of theoretical concepts in the music, while the notion of grouping discussed by a type-setter involves paging, bar lines and other print level information. Yet, all those grouping notions share one property: they combine independent subparts along a time scale. We call this property: Temporal Combination. The following figure tries to convey the intuitive idea of temporal combination.

**FIGURE 1 - Temporal combination:**

<table>
<thead>
<tr>
<th>Time scale of composite piece:</th>
<th>MS</th>
<th>start</th>
<th>0</th>
<th>25</th>
<th>50</th>
<th>clip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time scales of subparts:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS₂</td>
<td>start</td>
<td>0</td>
<td></td>
<td>clip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS₀</td>
<td>start</td>
<td>clip</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS₁</td>
<td>0</td>
<td>start</td>
<td>clip</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MS₀</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This picture describes a structured view of a piece that consists of three parts: part one, called MS₀, that starts at time point -10 (an introduction), part two, called MS₂, that starts at -1, and part three, called MS₁, that starts at 20. As can be seen in the figure, the components of the structure have their own time lines, with their own zero (beginning), and start and clip points. The temporal combination operation takes several such structured pieces and combines them along the time line of the composite piece.

The simplest, elementary, structured music piece is an occurrence of a sound. A composite structured music piece is a collection of time stamped structured pieces. A structured piece has its own time line. As elementary piece has a single occurrence of a sound object along its time line: in its beginning (zero); the occurrence lasts up to the time point specified by the duration of the sound object. In a composite piece, the components, which are, themselves, structured music pieces (elementary or composite), have their own time lines. The time stamp of a component denotes the displacement of its beginning from the beginning of the structured piece. Note that a piece does not, necessarily, start in its beginning. There might be an introduction or a delay.

In the Twelve Tone System, sounds are described as the combination of two properties: pitch and duration. Therefore, an elementary music piece can be viewed as consisting of a "durationless sound object", associated with a duration. The durationless sound object is the theoretical notion of a sound, stripped from its temporal
characteristics, i.e., when and how long it occurs. This view can be extended to any music in which the duration property of a sound can be distinguished from its other properties.

The following figures describe, graphically, the structured view of several known musical forms.

1) **FIGURE 2** - A 2 voices round:

   ![Diagram](image1)

   Components:
   - MS: start  clip
   - MS: start  clip

   Note that the components are the same, but their time scales are differently displaced.

2) **FIGURE 3** - A two part melody with an introduction

   ![Diagram](image2)

   Components:
   - INTRODUCTION: start  clip
   - PART1: start  clip
   - PART2: start  clip

3) **FIGURE 4** - A 4-part choir that starts with a delay:

   ![Diagram](image3)

   Voices:
   - VOICE1: start  clip
   - VOICE2: start  clip
   - VOICE3: start  clip
   - VOICE4: start  clip

The structured view of music does not enforce structuring of the music. On one extreme, one is free to assign music a MIDI like description as a stream of time stamped sound objects; on the other extreme, one can strive for an account of rhythmic and music patterns and their inter-relationships. For the same music piece, many structured views might make sense, depending on one’s point of view. All different structured views of a piece share the same denotation: the MIDI like, “flat” description.

3. Music Structures

In this section we briefly describe a representation language, called Music Structures, developed to account for the structured view of music described in the previous section. For a more detailed description, consult [Halaban 1988b]. A music structure describes one structured view of a music piece, and we distinguish between elementary
to non-elementary music structures. The primitives from which music structures are
constructed stand for durationless sound objects and time points.

The time primitives are the real (or rational) numbers. Any expression that evalu-
ates to a real number can be considered as denoting a time point. Therefore, a time-
stamp for a component or the duration of an elementary piece can be described by
numbers or by expressions like \(a + m\), or "duration(MS)" where MS is a given music
structure, and duration is a function that computes the durations of music structures.

The sound object primitives also depend on the music domain. In essence they can be
any symbolic representation for durationless sound objects. In the Twelve Tones Sys-
tem, where the conventional notation accounts only for the pitch and duration aspects
of a sound, the sound object primitives would represent conventional note-names such
as "G# in the 2nd octave (2G#)". In this domain, music structures are called Twelve
Tones Strings.

3.1. Elementary Music Structures

An elementary music structure is a pair \([p,d]\), that associates a term \(p\) denoting a
sound object primitive, with a term \(d\), that denotes a non-negative time point. The
term \(d\) is called the duration of \([p,d]\). An elementary music structure represents an
occurrence of a sound object for a "length of time" denoted by its duration. Its time
line includes a single occurrence that starts at time point \(0\) (the beginning), and ends
at the time point denoted by its duration (note that the duration actually denotes a time
interval).

Some examples of elementary music structures in the Twelve Tones System:

1. FIGURE 5 - Representation of notes and rests as elementary music structures
   (TTS-o):

   \[ p_1 = [\text{G}, \frac{1}{4}] \text{, } p_2 = [\text{C#}, \frac{1}{4}] \text{, } p_3 = [\text{G}, \frac{1}{4}] \text{, } p_4 = [\text{REST}, \frac{1}{2}] \]

2. Non explicit duration component:

   \([\text{G}E_b], \text{duration (MS)}]\]

   where MS is a music structure. This elementary music structure denotes an
occurrence of EB in the 0 octave, that lasts for a period of time equal to the dura-
tion of a piece denoted by the music structure MS.

3.2. Composite Music Structures

Recall that a composite structured music piece is a combination of structured
music pieces, each associated with a time stamp describing the displacement of its
beginning from the composite piece's beginning. The syntax for composite music struc-
tures is therefore straightforward: list notation and a time-stamp operator are the only
requirements. We write ms\(\text{st}\) to indicate that music structure ms is time-stamped with
time term t; ms\(\text{st}\) is called an occurrence.
A composite music structure is a list of occurrences. Here, we borrow the syntax of LISP. The composite music structure consisting of the occurrences \( m_i \) for \( 1 \leq i \leq n \), can be written either as \((m_1, m_2, \ldots, m_n, {}^{\text{nIL}})\) or more conveniently as \((\text{verty} m_1 m_2 \ldots m_n)\). The {} symbol is called the musical concatenation operator; it is the basic music structures operator.

**Examples**

1. The structured piece described in **Figure 1** in section 3.1, can be represented by a music structure, MS, as follows
   \[
   (MS_1@-10 \quad MS_2@-1 \quad MS_3@20)
   \]
   where \( MS_1, MS_2, MS_3 \) represent the three parts of the piece according to this view. Suppose now that \( MS_1 \) represents a structured piece consisting of an introduction whose beginning coincides with that of \( MS_1 \), and is followed by one part whose beginning coincides with that of \( MS_1 \). Then \( MS_1 \) is
   \[
   (\text{INTRODUCTION@-10} \quad MS_1@0)
   \]
   and the music structure MS is
   \[
   (\text{INTRODUCTION@-10} \quad MS_1@0 \quad MS_2@-10 \quad MS_2@-1 \quad MS_3@50)
   \]

2. A music structure describing a two voice round as in **Figure 2**, would be
   \[
   (MS@0 \quad MS@d), \quad d > 0
   \]

3. A music structure for a two part melody with an introduction, as in **Figure 3**, would be
   \[
   (\text{INTRODUCTION@d} \quad \text{PART}@0 \quad \text{PART}@\text{duration(PART)})
   \]

4. A music structure for a four part choir that starts with a delay, as in **Figure 4**, would be
   \[
   (\text{VOICE}@0 \quad \text{VOICE}@0 \quad \text{VOICE}@0 \quad \text{VOICE}@0)
   \]

Music structures can be generated by a wide range of music structures' operators. Two important operators are the horizontal and vertical concatenations. Horizontal concatenation, denoted by \("-\)\", means "followed immediately by". Vertical concatenation, denoted by \("\mid\)\", means "starts simultaneously with". Using these operators, a melody, consisting of the notes (with durations) \( e_1, \ldots, e_n \), can be described by:

\[
e_1 \mid e_2 \mid \ldots \mid e_n \quad \text{(or, for short:} \quad \frac{n}{m} \mid e_i \quad \text{)};
\]

A chord consisting of these notes would be:

\[
e_1 \mid e_2 \mid \ldots \mid e_n \quad \text{(or, for short:} \quad \frac{n}{m} \mid e_i \quad \text{)};
\]

A harmonic sentence, that is, a sequence of chords, would be:

\[
\frac{k}{m} \mid \frac{n}{l} \mid P_i \quad \text{.}
\]
a polyphonic sentence, that is, a simultaneous combination of melodies, would be:

\[ \sum_{i=1}^{n} P_i \]

Another special operator, that translates every music structure into a flat list of time stamped elementary music structures, is the flatten operator. The flat list can be considered as the canonical form of the music structure: It is shared by all music structures describing the same piece. In play-back time, a music structure should first be flattened into its canonical form. More operators are described in [Balaban 1988c].

The music structures language can be further strengthened by augmenting it with variables, user defined functions (λ-abstraction), and a naming mechanism. The inclusion of variables would allow for the description of partially specified music structures, musical patterns, durationless music structures, and rhythmic patterns. The naming mechanism is already used in the music workstation.

3.3. Atributed Music Structures

The music structures described so far can account for hierarchical organization of the sound objects. But a music score includes additional information, like, composer name, piece name, key signature, performance instructions, etc. This information can be viewed as a set of attributes, or attribute-value pairs, where an attribute is a name, and a value is an expression whose type is determined by the attribute. We use the term extended music structure to refer to the combination of a music structure and attributes' information. For example, a formula over a music structure MS would yield the extended music structure

\[ \langle MS, (formulas) \rangle \]

a phrase by the name of "First-motif", described under the key signature G♯, and played legato and with crescendo, would yield the extended music structure

\[ \langle MS, (\text{name, First-motif, (key-g#, G♯), legato, (dynamics, crescendo)}) \rangle. \]

Extended music structures can include also theoretical information such as the scale of a piece, chord analysis, etc.

4. A Music Structures' Based Workstation

The music workstation has two major parts: the back-end database and the user interface. The overall structure is described in the following diagram:
The music structures based workstation is currently under development in COMMON LISP, on a SUN 3/50 machine. LISP was selected due to its power in symbolic manipulation. The plan is to move the system, later on, to a PC environment, where a MIDI interface to keyboards can be used. For real-time applications, translation to a more efficient language (like C) might be considered. For a detailed description of the various components of the workstation, and current implementation status, consult the full paper ([Balaban 1987b]).

5. Conclusion

In this paper we have described the initial steps in building a music workstation, that is unique in its ability to account for multiple hierarchical views of music. The major development is done on a SUN machine, and the first planned "test site" is the research system where it was "born": the CSM system ([Balaban 1987a]). It is also planned to experiment with a non Twelve Tones domain. The environment "drawn" by [Uppenheim 1987] seems appropriate since it is also based on hierarchical structuring over a time line, using a graphical interface.

For more practical purposes like serving as a compositional tool, instructional aid, or for type-setting, a PC environment is planned. Here we would like to collaborate with music practitioners (for example, to try to teach a music course, like Harmony, where all music would run through the music workstation). Such collaboration would provide feedback from potential users, and, we believe, would emphasize the role of the temporal combination operators, and multiple hierarchical views, in music description.

Another direction would be to develop a print level extension, for typesetting purposes. This extension would require, first, study of the appropriate print level hierarchy (page organization into staves and measures as in [Gourlay 1986]), and development of algorithms that would transform music structures into the print level hierarchy.

63

ICMC Proceedings 1992
Acknowledgements

Peter Bielikarz and Dan Rowntrentz participated in the early design of the music workstation. Many ideas described in this paper were first developed in common discussions. I thank them both.

References


