

Music Representation Issues: A Position Paper *

Roger B. Dannenberg
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Designing a good representation for music is challenging because there are few if any generalizations about music structure that are universal. Rather than study musical conventions, designers of music representations should focus on the extremes of musical form. A representation that generalizes to new musical forms and notation is likely to provide elegant representations for more conventional music. On the other hand, a representation based on convention will be of limited use, and could be harmful by discouraging further development.

Introduction

Music representation is acknowledged to be a difficult design problem. One way to approach design problems is to look for the areas that are most difficult or that impose many constraints on the rest of the design. By nailing down the hardest and most constraining aspects of the design first, one will be more likely to find a design that satisfies all requirements.

In this spirit, I propose to look at some of the difficult aspects of music representation. In the following sections, I will briefly discuss algorithmic vs. declarative specifications, beat time vs. absolute time, hierarchy and multiple hierarchies, and multiple views and part extraction.

Algorithmic vs. Declarative Style

There has been great interest in specifying music by procedure as opposed to descriptive data. This trend has roots in traditional music in the form of repeats (iteration), first and second endings (conditional execution), codas (goto), figured bass (macros), and canons (higher level transformations). There are also precedents in other forms of representation such as VLSI and typesetting, notably in the PostScript language. In spite of these precedents, my position is that algorithmic or procedural descriptions are a bad approach to the representation problem because of the difficulty of manipulating these descriptions. PostScript is a good analogy -- there are no programs that manipulate arbitrary PostScript programs (other than ascii text editors). Subsets of PostScript such as "Embedded Postscript" have been developed to allow simple operations such as merging documents or inserting graphical figures into text documents. Embedded PostScript is a descriptive language that encapsulates procedural PostScript fragments.

*Published as: Roger B. Dannenberg, "Music Representation: A Position Paper," in *Proceedings of the 1989 International Computer Music Conference*, Kologne, Germany, October 1989. Computer Music Association, 1989. pp. 73-75.

Time

Time, the most essential ingredient of music, is difficult to represent because there are so many competing "theories" about time. In some music, time is measured in beats which can be thought of as a monotonically increasing function of time. In other music, beats do not exist and absolute time references are more appropriate. Some music combines several tempi, i.e. there are several simultaneous time bases. The tempi may or may not be rationally related and may or may not be constant.

In many cases, particularly when music is notated, the timing relationships among events must be explicit even when the events are specified in different time systems. For example, the horizontal position of a printed note is related to absolute time rather than beat time when multiple tempi are in effect. My conclusion from this observation is that it must be possible to derive a real time from an event representation.

A corollary is that using exact rational number arithmetic (where one keeps track of integer numerators and denominators as opposed to fixed- or floating-point representations) is much more difficult than it sounds. It is easy to construct examples where denominators require more than 64 bits if multiple tempi are allowed.

A representation must therefore deal with real time and beat time. The two are related by tempo, but the relation is complicated by the fact that tempi may be continuously variable and there may be multiple tempi affecting different musical lines. Composers do not always write theoretically correct music; that is, a measure of four beats may contain five quarter notes. Also, performers do not always play notes precisely, nor do composers necessarily want performances that adhere strictly to the notation. Performance timing should also be considered in a representation. These are the sorts of considerations involving time that must be accounted for in any representation system.

Hierarchy

Hierarchy is practically a buzz-word of music representation. Hierarchy has a place in most representation schemes, and the desire to cope with hierarchy generates many of the interesting representation problems. Simple hierarchies have a long tradition in computer science, and linear representations such as SGML pull in this direction. It is my position that music is better represented by multiple hierarchies, that is, a number of simultaneous but distinct simple hierarchies.

For example, I could describe a traditional piece of music in a horizontal hierarchy by breaking it up into movements, sections, phrases, measures, beats, and so on. I could describe the same piece in a vertical hierarchy by breaking it up into families (strings, winds, percussion), instruments (violins, trumpets), parts (first violin, second violin), and notes of a chord (on piano, or double stops on strings). Which of these is correct? The answer is both. One can imagine many other potentially useful hierarchies. My conclusion is that a music representation scheme should support multiple hierarchies.

Multiple Views

The concept of orchestral score and parts is particular to music, but the more abstract concept of multiple views is more universal. The term view comes by analogy to the physical world. Imagine taking a picture of something from different angles and under different lighting conditions. The photographs will all be different even though the object photographed is the same. We can think of parts, printed full scores, displayed bit-mapped images, midi files, and synthesized sound as possible views of a piece of music.

Should views be a part of the representation? One is inclined to say "no, a view is derived from the representation, but is not part of it". Unfortunately, this position is not

viable. It is known that complete automation of music typography is an unsolved problem, and therefore a representation that supports high-quality printing must have a place for typographical layout information. Typographical information for parts must be present along with information for the main score. Thus, a representation that supports high-quality typography must support multiple views at least insofar as parts go.

Current computer-based music notation systems do not support parts. In many cases, there are facilities for separating a score into parts, but these parts become scores in their own right. Changes to the score are not propagated to the parts, and manual corrections to the layout of a part will be lost if the part is extracted anew from the score.

My position is that multiple views should be supported fully by a representation. What does this mean? Many schemes are possible, but it should at least be possible to associate attributes of events with particular views. For example, there might be a position attribute with one value for the score view and another value for the violin-1-part view.

Further Reading

Donald Byrd's (1984) thesis is essential reading and a source of many of my comments here. Read the thesis not so much for its discussion of how to automate music notation but for examples and discussion of the difficulty of "fully automatic" notation. While Byrd limits his examples to "major" composers and publishers, contemporary music scores are a rich source of examples that a designer should consider.

The *Directory of Computer Assisted Research in Musicology* is published annually by the Center for Computer Assisted Research in the Humanities, Menlo Park, CA. The editors have done an extraordinary job of assembling examples and descriptions of music representation and notation software.

An example of a notation system supporting multiple views and multiple hierarchies was described in a previous conference (Dannenberg 1986).

Conclusions

I have presented four tough problems for music representation systems. I do not know of any commercial systems that handle any of the latter three, and I think this indicates the lack of maturity of the field in general. It also indicates that there is commercial value in solving the easier representation problems and ignoring the rest. It is only a matter of time, however, before more comprehensive systems are devised and these problems become important issues. We can guide future developments in music software by providing an understanding of the important issues and by formulating comprehensive representation systems that will provide a base upon which others can develop.

References

- Byrd, D. 1984. *Music Notation by Computer*. Department of Computer Science, Indiana University. (Ph.D. Thesis).
- Dannenberg, R. 1985. "A Structure for Representing, Displaying, and Editing Music." *Proceedings of the 1985 International Computer Music Conference*. Computer Music Association. pp. 153-160.