Higher-level Composition Control in Music Sketcher: Modifiers and Smart Harmony

Steven Abrams, Daniel V. Oppenheim, Don Pazel, James Wright

{ abrams, music, pazel, jwright @watson.ibm.com }

Computer Music Center
T. J. Watson Research Center, IBM
P.O. Box 218, Yorktown Heights, NY 10598
www.research.ibm.com/music

There is a fundamental tension between working with high-level abstractions which often provide intuitive conceptual manipulations but hide details, and working with low-level data which, while quite powerful, can also be quite tedious. To help balance this tension, we have been developing technologies and higher-level abstractions designed to enhance the composer's ability to create music, focusing on the compositional concepts of shape, structure, and tension-release. This paper discusses our ongoing efforts in this area, as demonstrated in the Music Sketcher program, with particular focus on the Smart Harmony and Modifier technologies.

1. Introduction

The typical user interface in many commercial music applications models machinery in a recording studio, such as a multi-track recorder and mixer. This model has little to do with the process of creating music that has been evolving for over a millennium by Western composers. Computer languages for composing music that are more commonly developed in academic environments share a similar problem. In these languages the conceptual model is often a combination of some structural aspects of music as modeled by a musical score, with a focus on tools and processes that are often borrowed from computer science. Here the connection to traditional compositional process is tenuous, too.

At IBM Research's Computer Music Center (http://www.research.ibm.com/music) we are engaged in a series of projects that revolve around music, art and creativity, with a goal of helping people to amplify their natural musical creativity. In particular we are beginning to investigate some of the concepts used by composers in their creative process, starting with structure, shape, and tension-release. We feel that by focusing on concepts, rather than on machinery that happens to be available, we will be able to develop user interfaces, tools, and technologies that are more meaningful to composers. This paper will focus on three novel mechanisms that we developed for dealing with these initial concepts: blocks, modifiers, and Smart Harmony. "Blocks" is our initial attempt at a technique for manipulating a composition's structural aspects. "Modifiers" allow reshaping aspects of the music at different structural levels, ranging from subtle expressive nuances to significant reworking of compositional materials [see Oppenheim 92, 96, Wright et. al. 97]. "Smart Harmony" models aspects of tonality – a mechanism used in Western music to create sensations of tension and release – and supports the creation of harmonic frameworks in a composition. These are all implemented in our Music Sketcher application, available at http://www.research.ibm.com/music.

Music Sketcher is a vehicle we are developing for experimenting, generating ideas, and exploring the power of our new music technologies. It draws on our experience from DMIX [Oppenheim 96] and builds upon our previous work with the CyberBand program [Wright, Oppenheim et al 1997]. Music Sketcher is unique in that it combines a model of tonal music with a system of high-level modifiers that supports a variety of sophisticated manipulations, all working together to maintain a desired musical context. Music is represented using a three-level content hierarchy (see Figure 2): a single ScoreSheet contains a set of ScoreParts, each containing one or more ScoreBlocks. The ScoreSheet represents the entire composition in a fairly conventional manner: time proceeds from left to right, and concurrent Parts are stacked vertically. A ScoreBlock (or *Block*) contains an ordered set of notes and other events, often corresponding to a musical phrase, motif or "riff" (a *Riff Block*) or even a chord sequence (a *Harmony Block*). A Block could also contain entire sections or nested compositions: the underlying representation supports hierarchies of arbitrary depth (Music Sketcher does not currently use this capability).

A parallel modifier hierarchy mirrors the multi-level content hierarchy and provides an intuitive mechanism for creating musical transformations. Smart Harmony works in tandem with pitch modifiers and ensures that pitches will always maintain their correct harmonic function. This opens the door to many new kinds of musical transformations we feel are useful. For example, any riff can be placed in any harmonic context and adapt to it while maintaining its essential melodic and rhythmic characteristics. In addition, any melodic line can be reshaped while ensuring that the new pitches always function correctly within the given harmonic context. The remainder of this paper will discuss Modifiers and Smart Harmony in more detail

2. Modifiers

Modifiers are real time transformation algorithms that change aspects of musical events such as pitch, dynamics, onset and duration. They can operate at any hierarchical level of the music representation, and may be used to make both subtle and radical changes to a musical entity or set of entities. For example, a pitch modifier might be used to reshape a melodic line (see Figure 2 and staffs 2 and 3 in Figure 3). A duration modifier can be applied to a violin part, ensuring that all musical fragments placed in that part are rendered *staccato* or *legato* in varying degrees, according to a particular articulation pattern (see staff 4 in Figure 3). Modifiers may be combined to create a single complex effect. For example, the articulation and phrasing of a particular motif can be completely reshaped by combining onset, velocity and duration modifiers.

The multi-level content hierarchy is mirrored by a parallel modifier hierarchy. Modifiers attached to a content element at a given level affect both that content element and any lower-level content elements nested within the parent element. The explicit, multi-level representation of content modifications has at least four major benefits: (1) A desired musical result can be easily described. (2) Modifiers are *late-bound* (effects occur during playback). Thus, modifiers themselves can be altered directly and dynamically, both during playback (e.g. for performance purposes) and as part of editing the actual composition. (3) Modifiers can be *inherited* simply by placing a content element in a given location within the dual content-modifier hierarchy. All modifiers for a given location (e.g. bars 5-9 in the Lead Part) are applied automatically to any block(s) placed in that location. (4) It is easy to understand and manage interdependencies between different modifiers, both within a given level and across multiple levels.

In Music Sketcher, Modifiers are currently modeled as graphical curves which change some aspect of the music over time. By default, the length of a modifier curve stretches as needed to fit the duration of the associated block or score portion of the score sheet. However, you can also create cyclical effects, by setting the modifier to repeat for a specific duration (see both Modifiers in Figure 2).

Within Music Sketcher, using a modifier is fairly straightforward (see Figure 2): (1) Select the aspect of music you want to affect (e.g. pitch, duration, velocity, onset); (2) draw a curve that describes the desired change over time; (3) set the vertical range of the shape (e.g. will a pitch curve span an interval of a third, or five octaves); (4) select a combination algorithm to control how the graphical curve is applied to the current value(s) of the musical events being processed. When several modifiers are applied to the same musical aspect (such as pitch), you can specify the order in which each modifier is applied. Individual modifiers may also be toggled on and off.

As previously noted, when pitch modifiers are used in conjunction with Smart Harmony then new and useful musical transformations are made possible. Rather than merely a diatonic or chromatic transposition, pitches can be manipulated in various ways while maintaining conformity with the underlying harmonic context.

3. Smart Harmony

Smart harmony provides a means for transforming the pitches of a block, making them conform to a given specification of harmony. Techniques used in Smart Harmony were based on work from DMIX [Oppenheim, '96] that elaborated a representation of Pachet [Pachet, '93]. Harmony is specified by a (tonic, modality, chord-root, chord-type) tuple, called a *harmonic context*. These are arranged in *Harmony Blocks* that are placed in the *Harmony Part*, which provides the harmonic framework for the composition. Smart Harmony supports two types of transformations: chord-changes and pitch-shifts.

When a riff is played against a harmony part, the chord-change transformation occurs. Smart harmony alters the pitches to work musically in the designated harmony while retaining the riff's essential character – even if the new harmony is radically different the riff's original harmony.

This process is not a simple key transformation. Each pitch in the riff contains a description of its harmonic function, indicating the pitch's role within its original chord and key. Examples include whether or not it is a chord pitch, a passing scale tone, a chromatically altered scale-tone, etc. When a pitch is rendered against a given harmonic context, smart harmony finds a pitch near the original pitch that has a compatible function in the new harmonic context. The preservation of harmonic function is non-linear, and is especially noticeable in chord changes, wherein different pitches are shifted by different amounts. The end result is a riff sounding much like the original but noticeably in the new harmony.

The pitch-shift provides for pitches to be moved higher or lower by a specified chromatic interval. However, instead of precision adjustments to the interval, the pitches are transformed to pitches, "close by" in the interval, which preserve their harmonic function. This transformation when applied in conjunction with pitch modifiers provides a melodic reshaping capability that retains harmonic conformance. As used in the MusicSketcher interface, one can literally, draw a new pitch line for a melody and expect a reasonable audio rendering of the melody closely adapted to that line, but with a preserved harmony.

4. Harmony Builder

The harmony builder is a visual tool in MusicSketcher that is used to compose sequences of harmonic contexts, or progressions (the *Harmony Blocks*), which are inserted in a MusicSketcher ScoreSheet's harmony track. The harmony builder is invoked directly from MusicSketcher and interacts with the ScoreSheet, allowing blocks to be transferred to and from the ScoreSheet's harmony track

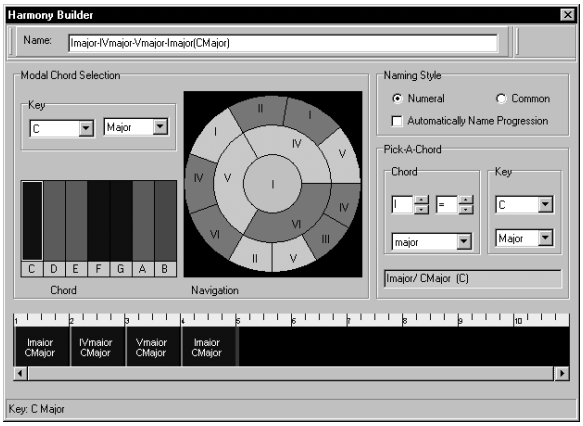

Figure 1: Harmony Builder

One uses the harmony builder in the following manner. After selecting a key, a user selects chords from one of several visuals and inserts them into a progression assembly region (shown at the bottom of **Error! Reference source not found.**). The user places these chords in succession into the progression and adjusts the onset and duration of each in beats. When finished, the progression is simply dragged from the assembly area and dropped onto the harmony track at an appropriate position.

Central to the harmony builder is a chord navigation visual, which assists the user in building chord progressions that make sense. Concentric regions, each partitioned into sectors provide a guide map for chord succession. For example, starting at the cord in the center, the next successor chords lie in the next outer concentric region. Lighter colored sectors indicate "more likely" successors than sectors colored darker. Continuing, successors to these are found in the outer concentric region similarly. From there, the diagram may be re-centered to a different chord, and the process proceeds. The current chord succession rules are based on [Piston '87] and are limited to diatonic triads. We are developing a more flexible mechanism for dynamically making alternative harmonic rule bases available.

The harmony builder also assists in modulating to related keys. Right-clicking on a chord in the assembly area brings up a list of alternative representations of that chord in other keys, allowing that chord to be used as a pivot for modulating to a new key.

The harmony builder also includes a pick-a-chord feature allowing selection of non-diatonic chords, including 7^{th}, 9^{th}, 11^{th}, and 13^{th} chords with augmented or diminished notes and built on arbitrary scale degree. Chords specified in this manner may also be dropped into the assembly area.

5. Conclusion

Although it is not yet a complete compositional tool, Music Sketcher demonstrates the novelty and utility of these technologies. We intend to continue developing these technologies, adding to the functionality and the usability, as we better understand how higher-level operations and abstractions can truly assist in the creative process.

6. References

Oppenheim, D. (1992). "Compositional Tools for Adding Expression to Music." *Proceedings of the ICMC*, San Jose, California.

Oppenheim, D. (1996). "DMIX—A Multi Faceted Environment for Composing and Performing Computer." Computers and Mathematics with Applications, 32:1, pp. 117-135, 1996.

Oppenheim, D., Wright, J. (1996). "Towards a Framework for Handling Musical Expression." *Proceedings of the ICMC*, Hong Kong.

Pachet, F. (1993). "An Object-Oriented Representation of Pitch-Classes, Intervals, Scales and Chord: The basic MusES" (revised and extended version), LAFORIA, Internal Report 93/38, November 1993.

Piston, W. (1989). "Harmony", WW Norton & Co.

Sadai, Y. (1980) "Harmony in its Systemic and Phnomenological Aspects." Yanetz Press. Jerusalem, Israel.

Wright, J., Oppenheim, D. (1997). "CyberBand: A 'Hands-On' Music Composition Program" *Proceedings of the ICMC*, Thessaloniki.

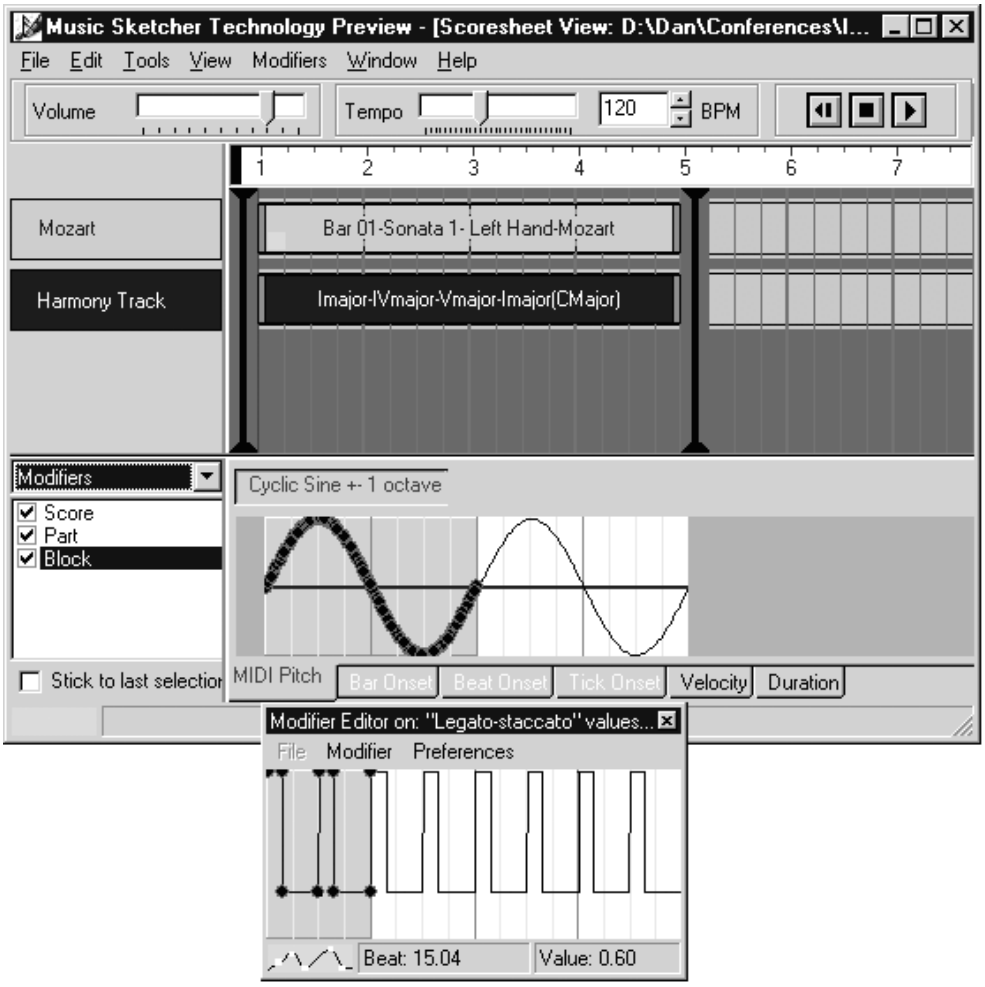

Figure 2: The Music Sketcher application

The Mozart Riff is stretched over four bars in the top part. The Harmony track contains the harmonic progression I-IV-V-I, created with the Harmony Builder in Figure 1. The Modifier view displays the "Cyclic Sine +- 1 octave" that applies a sine shape to the pitches of the Mozart riff every two bars. An additional Modifier "Legato Staccato" is applied to duration.

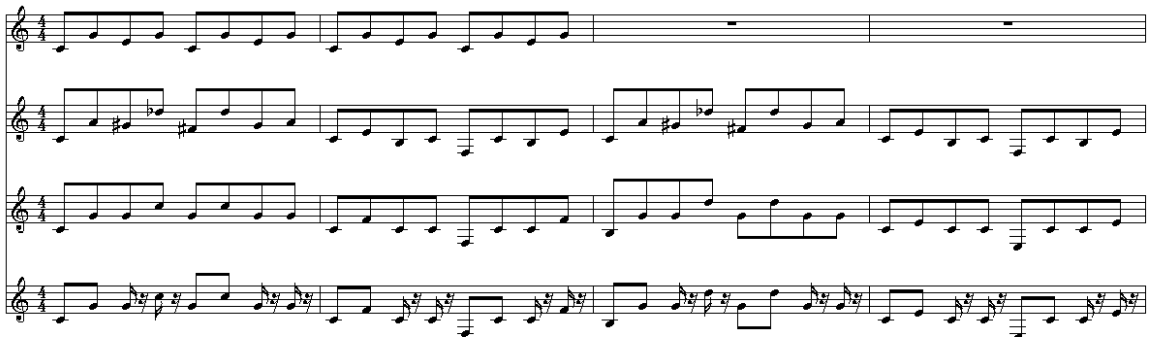

Figure 3: Music as processed by Modifiers in Music Sketcher

1. Staff 1: the original Mozart riff, repeating two times.
2. Staff 2: the Mozart Riff stretched over four bars. A cyclic pitch modifier is applied repeating its shape every two bars. The range of the resulting transposition is one octave in each direction. Smart Harmony is disabled.
3. Staff 3: the same, but now Smart Harmony is applied. Note that pitches are conformed to the underlying harmonic progression (I-IV-V-I). Since all notes in the Mozart riff are chord tones, pitches are shifted to the appropriate chord tones of the underlying progression.
4. Staff 4: the same, but here the cyclic Modifier "Legato Staccato" is applied to duration, affecting each bar.

