A Theory of Poetry as Music and Its Exploration through a Computer Aid to Composition

Fred Lerdahl, USA and Xavier Chabot, IRCAM
31 Rue Saint-Merri, 75004 Paris, France

Abstract

Lerdahl and Halle (in press) draws an extended mapping between between the sound structure of poetry and the hierarchical representations in Lerdahl and Jackendoff's (1983) theory of music. These parallels suggest transferences between the two arts by means of digital technology. Our first objective is to effect a step-by-step transformation of a poetic text into musical sounds while preserving perceptually the structures represented by the theory. This research can then serve as a platform for musical composition.

To accomplish these goals we have designed and implemented a computer aid to composition within the framework of the PatchWork environment on the Macintosh. This paper presents the current status of the project through a live demonstration. The theory is outlined, the environment is presented, a sampled line from a Shakespeare sonnet is manipulated from PatchWork, and Chant synthesis is performed under the control of the same environment.

The Music-Poetry Theory

Lerdahl and Jackendoff (1983) models musical listening by means of four hierarchical components: grouping structure, metrical structure, time-span reduction, and prolongational reduction. Aspects of linguistic structure have a comparable organization, as the following summary outlines.

(1) The prosodic hierarchy (which is distinct from the syntactic segmentation) is a hierarchical grouping of spoken segments, comprising the units of the syllable, the word, the clitic phrase, the phonological phrase, the intonational phrase, and the utterance (Hayes 1989).

(2) Within each of the units from word to utterance, certain syllables receive more stress than others. In most poetry the stresses occur at regular enough intervals to induce a sense of meter. Though rarely isochronous, a poetic meter has the same formal structure as its musical counterpart, a leveled grid expressing periodic patterns of strong and weak beats.

(3) Recent work in linguistic theory (Halle & Vergnaud 1987) notes phonological stress by a combined grid and grouping notation that can be represented by the musical time-span component, which assigns pitch stability within rhythmic structure. Similarly, the linguistic time-span analysis represents syllabic prominence within the prosodic hierarchy and in relation to the poetic meter.

(4) An important feature of poetry is the pattern of full or partial recurrences of syllables, whether by rhyme, alliteration, or assonance. These patterns resemble the departure and return of pitch events described by prolongational reduction, but in the timbral domain. The resulting analysis is like the timbral prolongational structures explored in Lerdahl (1987).

(5) A more pitch-like analogy to prolongational reduction concerns the melodic contour of an utterance. Speech contour in turn depends on the meaning conveyed (Bollinger 1986); in particular, contour peaks are governed by content words as projected by the prosodic hierarchy. Contour analysis thus becomes a kind of semantic prolongational structure. The following summarizes these componential mappings between music and poetry.

<table>
<thead>
<tr>
<th>Music</th>
<th>Poetry</th>
</tr>
</thead>
<tbody>
<tr>
<td>grouping</td>
<td>prosodic hierarchy</td>
</tr>
<tr>
<td>meter</td>
<td>meter</td>
</tr>
<tr>
<td>time-span reduction</td>
<td>phonological stress hierarchy</td>
</tr>
<tr>
<td>timbral prolongation</td>
<td>phonological prolongational structure</td>
</tr>
<tr>
<td>pitch prolongation</td>
<td>semantic prolongational structure</td>
</tr>
</tbody>
</table>

ICMC 328
We pass over specific derivations and consider only their outputs in the form of notations as in the figure below. The text is the first line of Shakespeare's Sonnet 29. The tree above the line models the semantic prolongational structure and the slurs beneath represent its phonological prolongational structure. The metrical structure is given by the dot grid. The prosodic hierarchy appears at the bottom as the time-span segmentation, with the time-span reduction notated by syllables at different levels.

Implementation with PatchWork

A computer-aided composition environment is a set of tools for the composer to deal with musical material at a precomposition stage where imagination and rigor compete for expression. Such an environment must provide an extensible set of tools not only to create material but to represent it, play it back, and modify it. The environment must be able to communicate with other programs for editing, performance, notation, and sound synthesis. It must allow the composer to reuse earlier stages of work for further development and to embed material within a structure at a larger hierarchical level. This point is essential if there is to be a relationship between material and form. PatchWork is an interactive environment for composition and synthesis control. The present version has been written in Common Lisp by Mikael Laurson, Camilo Rueda and Jacques Duthen (Laurson & Duthen, 1989, 1990). PatchWork allows the composer to create and manipulate musical material (scores, rhythms, envelopes, formal structures), to exchange this material with other environments such as MAX or a music notation program, to implement algorithms for manipulating musical objects and formal structures, and to control various synthesis models (MIDI synthesizers, sound, CHANT). PatchWork's basic paradigm is that of functional programming through graphic patches.

Patches are convenient for implementing functional trees that represent the structures assigned by the music-poetry theory, to perform actions at specific nodes in the structure, and to control sound playback and synthesis. The figure below shows the various parts of the PatchWork implementation of the music-poetry theory. All components are from the regular PatchWork environment except the tree node module.

The functional tree represents a particular hierarchical structure. The tree-collector module first evaluates the functional tree and applies a list of actions specified at each node to the sound synthesis model; then it schedules and generates instances of the sound model in an event sequence stored in and displayed by the music notation editor. The playback and synthesis commands are interpreted depending on the type of note objects collected in the score.

ICMC 329
The tree below once evaluated produces a functional expression such as:

\[(\text{amp} \ 60 \ (\text{pitch} + \text{-110} \ (\text{leaf} \ 0 \ \text{leaf} \ 1))) \ (\text{delay} \ 10 \ (\text{leaf} \ 2 \ \text{leaf} \ 3))\]

Choosing one action (pitch transposition, delay to change rhythm, amplitude) from the popup menu at a particular point in the tree displays the parameter values for this action at every node in the tree, and allows editing of this action parameter. In the example above, parameters for change in amplitude are displayed, but other actions (transposition and delay) appear in the expression because they have been set before. Actions from node to node are additive.

The Functional Tree in Action

For the demonstration, the functional tree is applied to two different types of sound objects: (1) sample objects that control the playback of samples in a sampling machine; (2) CHANT instrument objects that control CHANT synthesis (Barrière, et al., 1995). Concerning (1), the samples are from a reading of Sonnet 29 cut into syllables. The syllables are flangeted out in timing, pitch, and amplitude to establish a baseline from which subsequent modifications can be measured.

Acting on the sound objects through a functional tree ensures that the processing follows the rules assigned by the theory. A functional tree built from the semantic prolongational tree automatically assigns pitch contour, and a tree built from the metrical and time-span structures automatically assigns a durational sequence. When played with the sampled voice-objects using both trees at the same time, the result is a kind
of test of the theory. To the extent that the theory is correct, the phrase should sound natural.

Finally, an example with CHANT synthesis control is presented in the demonstration. The CHANT sound objects replace the sampled voice-objects as the sound input to the tree collector. The CHANT objects follow the sequence of vowels in the Shakespeare text, creating a melodic simulation of the sampled version.

The project was originally stimulated by the hybrid music-poetic compositions of Berio, Dodge, Lansky, and Ligeti. Its approach shares features with Pope (1990). The computer aid revives ideas in Lerdahl and Piotard (1986). In future developments, we envision controlling a Phase Vocoder (DePalle & Poirer 1991) processing voice sounds, so that the time and frequency domains can be treated separately. The CHANT objects need to be enriched with appropriate interpolated consonants. At some point research will give way to composition.

References


ICMC 331