

A Tool for Manipulating Expressive and Structural Hierarchies in Music (or, “T-R Trees in the MODE: A Tree Editor Based Loosely on Fred’s Theory”)

Stephen Travis Pope

Computer Music Journal, Nomad Computer Music Research,
CCRMA: Center for Computer Research in Music and Acoustics, Stanford University
P. O. Box 60632 Palo Alto, California 94306 USA
Email: stp@CCRMA.Stanford.edu

Abstract

The *T-R Trees* software system is a set of software tools for the graphical and programmatic manipulation of expressive and structural hierarchies in music composition. It is loosely based on the hierarchies described in Fred Lerdahl and Ray Jackendoof’s landmark book *A Generative Theory of Tonal Music*—weighted grouping and prolongational reduction trees (also called tension-relaxation or T-R trees). This article describes T-R tree derivation, editing, and application in score representation and management.

Introduction

“T-R Trees” is a software package for composition and interpretation of structured music. The basic representation used in the T-R trees system is a weighted hierarchical tree similar to the prolongational reduction trees of the generative theory, or to the prosodic stress trees used by linguists to notate the stress of spoken utterances. A T-R tree can in general be derived from any property of an event list (e.g., a melody or a spoken phrase), by ordering the events according to that property and generating a binary tree where node position and branch type are used to convey information about the division of the chosen property among the “leaf” events.

The application of the cross-mapping between metrical grouping, prolongational reduction, and prosodic stress trees is a very powerful technique for the representation and manipulation of several levels of expressive and structural information in scores. T-R trees can, for example, be derived from the phrasing of the words in a line of text, and then applied to—mapped onto, so to speak—the loudness or timbre of another musical structure.

The system described here is implemented as a new class library within the MODE (Musical Object Development Environment) (Pope 1991), an object-oriented Smalltalk-80 framework and tool kit for tasks related to music representation, performance and human-computer interfaces for musical activities.

A Generative Theory of Tonal Music

The theoretical basis of (well, inspiration for, at least), the T-R trees system is the “Generative Theory of Tonal Music” presented by Fred Lerdahl and Ray Jackendoof in their monumental book (Lerdahl and Jackendoof 1983) and a companion paper called *A Computer Aid to Composition* (Lerdahl and Potard 1982). The generative theory describes four primary kinds of structural trees,

along with sets of well-formedness and preference rules for deriving each type from a musical event surface. The hierarchies it formalizes are:

- grouping structure (hierarchy of event grouping);
- metrical structure (alternation of strong and weak beats);
- time-span reduction (pitch weighting according to strength); and
- prolongational reduction (harmonic and melodic tension/relaxation).

The primary structural abstraction of the T-R Trees system is a weighted binary tree based rather loosely on the prolongational reduction or tension-relaxation (T-R) tree of the generative theory. A T-R tree can in theory be derived from any property of an event list (e.g., a melody or a spoken phrase), by ordering the events according to that property and generating a binary tree where right- and left-branching nodes denote tensing and relaxing—exiting and entering more- or less-stable states—respectively.

The last chapter of *A Generative Theory* includes a section titled *A Deep Parallel between Music and Language*. The authors identify here the strong similarity between the trees derived from musical surfaces via time-span and prolongational reduction and the prosodic tree structures used by linguists to notate weak and strong syllables within utterances. The application of this cross-mapping became a central technique in the representation of the score for *Celebration*—a recent composition of the author’s realized using the system.

It is straightforward to extend the notation from T-R trees and prosodic stress trees to arbitrary weighted binary trees. First, one needs to note that prosodic stress trees are simply weighted binary trees where the weighting is mapped from verbal stress of syllables within an utterance. The four types of T-R trees described in the generative theory are not weighted *per se*, and the notation of the three types of branching is different, but one can easily assign weights to the quanta of each type of tree (e.g., strength of metrical grouping or closeness of pitches in time-span reduction).

The T-R Tree Framework in the MODE

T-R Trees is programmed as a collection of Smalltalk-80 Release4 class hierarchies that implement the basic representation of events, event-lists and trees, and the interactive editor and browser applications for composing using MODE and T-R Trees. In the style of the rest of MODE, the T-R Trees classes are constructed for easy re-use and customization. The abstract classes used for weighted trees and maps are easily extended for each specific scenario in a composition by the addition of new theories and new applications.

A *theory* consists of a tree node class—with knowledge of well-formedness rules for structures and optional behaviors for tree manipulation—and a map class—with methods for maintaining tree consistency by transforming nodes under certain tree operations. An *application* has a “perspective”—a graphical layout and spece/structure mapping algorithm—and one or more editor/browser/controller components for editing structures within the given perspective.

The provision of “hooks” for building new hierarchies and theories make the system quite flexible, and experiments ranging from the very low-level expressive domains up to managing more high-level structural elements have been undertaken.

Examples

In using the “MODE + T-R Trees” system, several different graphical and textual formats can be used for generating, managing and realizing a score. These include: graphical event list (i.e., melody or utterance) editors based on common pitch-time diagrams similar to common-practice western music notation; graphical editors for spatial location trajectories; programmed event generator scripts (i.e., procedural Smalltalk-80 algorithms describing “middle-level” musical structures (Pope 1989)); and graphical tools for the derivation, editing and application of T-R trees.

Two examples of one type of T-R Tree browser view are shown in Figures 1 and 2 below. The texts are two sentences from the poem *Moonlight Night (Ywe Ye, Yi Jr Di)* by the T’ang Dynasty Chinese poet Du Fu. The four list views along the top of the browser show sections, phrases, theories, and trees. The sections shown are the second movement of *Celebration*; within the selected section (*Lament*), one can see the eight lines of the poem. For each line, we have a set of tree types, as shown in the theories list; and there can be many trees within each theory, e.g., versions of one weighting tree.

For the left-hand example, you are looking at the “sharpness” tree of the “friday” version of the fourth line of the poem (“The moon is much brighter over my home-town tonight.”); this shows a strong dominance of “shr” and “syang” syllables. According to way the “sharpness” theory is implemented, changing the weights of this tree (by “dragging” a node in the tree with the mouse), would change the filter coefficients for the synthesis of the relevant syllables (thereby changing their “sharpness”). The right hand view, Figure 2, shows one “length” tree of the last sentence of the poem (“And the on-going war only makes matters worse.”), were the “nai” is being exaggerated in length. Editing this tree would change the relative durations of the syllables.

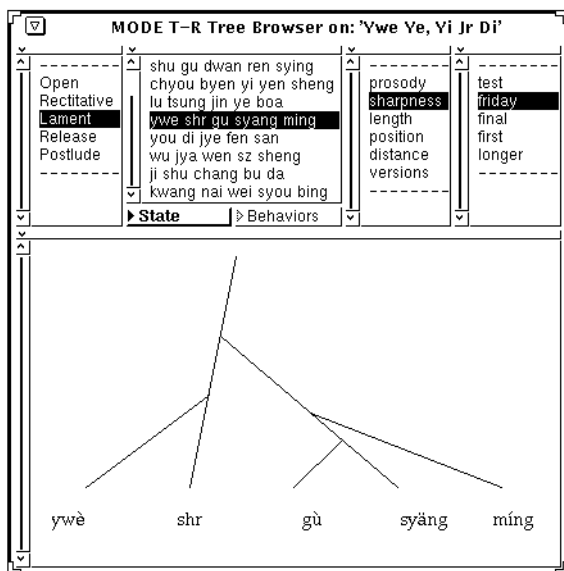


Figure 1

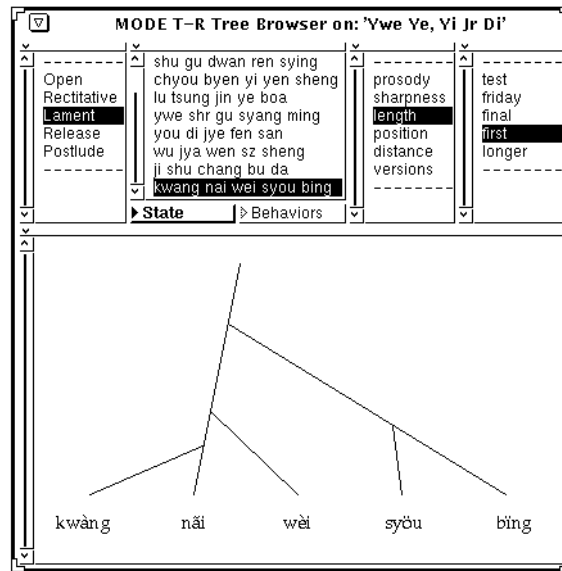


Figure 2

Figure 3 shows an older T-R tree editor of the type that was used for generating the vocal parts of the first movement of *Celebration, Kombination XI*. The figure on the left shows the prosodic stress tree for the given word, and the lower part shows its particles. (The German word *dunkelkammergespräche* means “dark-room discussions” and is one of the central words and T-R trees of the movement.) The view shown in Figure 4 is a location-time notation editor; stereo position is mapped onto the vertical axis with the top corresponding to the left channel and the bottom to the right channel. It shows a section where two left-right panning streams of “k” sounds are being generated and refined. In these streams, the left-right movement is controlled by Smalltalk-80 event generator code (not shown), and the selection of which “k” sound to use is made by mapping a T-R tree onto the rhythmic values based on the ordering of the four available sounds (labeled “k1” through “k4”).

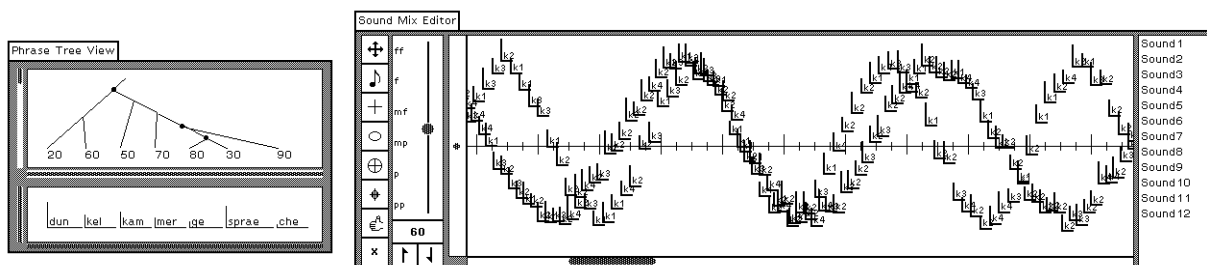


Figure 3

Figure 4

Conclusions and Directions

The T-R Trees system within the MODE framework is being used by the author for “real” composition and realization projects on Sun Microsystems SPARCstations, Apple Macintosh and other hardware platforms. The MODE system (including the current version of T-R Trees), is available to the public in Smalltalk-80 source code form via anonymous ftp from *ccrma.Stanford.edu*.

There are many missing features and their provision is an on-going project. The system interfaces to MIDI, DSP coprocessors and other I/O channels are still worthy of speed and robustness tuning. At present the amount of data that is generated by the version-handling components of the MODE’s event manager during a typical editing session is also a problem. Version-handling is normally turned off, making the use of many of the hypermedia facilities of the system impossible. As with the rest of the MODE, the development is being driven by the author’s own immediate compositional needs rather than the desire to make a perfect software engineering artifact.

References

- Lerdahl, F., and R. Jackendoof, A. 1983. *Generative Theory of Tonal Music*. Cambridge: MIT Press.
- Lerdahl, F., and Y. Potard. 1982. *A Computer Aid to Composition*. Paris: Rapport de l’IRCAM.
- Pope, S. T., 1991. “Introduction to MODE: The Musical Object Development Environment,” in S. T. Pope, ed. *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology*. Cambridge: MIT Press. pp. 83-106.
- Pope, S. T., 1989. “Modeling Musical Structures as EventGenerators,” *Proceedings of the 1989 International Computer Music Conference*. San Francisco: Computer Music Association. pp. 249-252

MODE T-R Tree Browser on: 'Ywe Ye, Yi Jr Di'

| | | | |
|---|---|---|---|
| Open Rectitative Lament Release Postlude | shu gu dwan ren sying chyou byen yi yen sheng lu tsung jin ye boa ywe shr gu syang ming you di jye fen san wu jya wen sz sheng ji shu chang bu da kwang nai wei syou bing | prosody sharpness length position distance versions | test friday final first longer |
|---|---|---|---|

▶ State | ▶ Behaviors

```
graph TD; Root[ ] --- ywe; Root --- Node1[ ]; Node1 --- shr; Node1 --- Node2[ ]; Node2 --- gu; Node2 --- Node3[ ]; Node3 --- syang; Node3 --- ming;
```

ywè shr gù syäng míng

MODE T-R Tree Browser on: 'Ywe Ye, Yi Jr Di'

| | | | |
|---|---|---|---|
| Open Rectitative Lament Release Postlude | shu gu dwan ren sying chyou byen yi yen sheng lu tsung jin ye boa ywe shr gu syang ming you di jye fen san wu jya wen sz sheng ji shu chang bu da kwang nai wei syou bing | prosody sharpness length position distance versions | test friday final first longer |
| State Behaviors | | | |

```
graph TD; Root[ ] --- Node1[ ]; Node1 --- kwàng[kwàng]; Node1 --- Node2[ ]; Node2 --- nāi[nāi]; Node2 --- Node3[ ]; Node3 --- wèi[wèi]; Node3 --- Node4[ ]; Node4 --- syöu[syöu]; Node4 --- bing[bing];
```