

# Producing *Kombination XI*: Using Modern Hardware and Software Systems for Composition

Stephen Travis Pope, Composer, Software Engineer and Editor  
Nomad OD/CMR, *Computer Music Journal*,  
Center for Computer Research in Music and Acoustics (CCRMA),  
Department of Music, Stanford University  
P. O. Box 60632 Palo Alto, California 94306 USA  
Electronic Mail: stp@CCRMA.Stanford.edu

## Abstract

This article discusses two topics related to the realization of my composition *Kombination XI: A Ritual Place for Live and Processed Voices*. These are the score's structure representation language and the software tools for manipulating it using graphical structure editors, and the process of realization using several different digital signal processing software and hardware systems. The reason for focusing on the first issue is the attempt to build a notation and set of software tools based on weighted trees that span the expressive and structural domains of music. The second topic is of interest as an example of the possibility of using several types of computer hardware and software in consort as one instrument. Numerous score and structure description and editing examples, and documentation of the realization process are presented.

## Introduction

This article discusses the application of state-of-the-art computer music and digital audio signal processing hardware and software in the realization of the composition *Kombination XI: A Ritual Place for Live and Processed Voices*. The piece is a setting of the poem *Kombination XI* by Helmut Heissenbüttel [1], and (in the style of the tape music of the 1950's and 1960's (*musique concrète*)), all of the sounds in the composition are derived from the voices of two speakers (one male and one female), reading the poem.

The score for the composition was expressed in terms of a set of weighted binary trees (called tension-relaxation trees for reasons that will become apparent), that were derived from the prosody of the text. These trees were applied to the organization of the musical materials of the piece on several levels.

The production process used Sun SPARCstation, Apple Macintosh, NeXT, and IMS Dyaxis computer hardware. The software included a phase vocoder program for voice signal processing, Smalltalk-80 software for score processing, the *cmusic* sound compiler language for sound editing and mixing, and the MacMix front-end software for the Dyaxis digital mixing system. The production was undertaken at the studios of the Vienna Music Academy, the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University, ParcPlace Systems, Inc., QuickSilver Studios, and my home studio.

This article will describe the process of composition and realization and present examples of the score formats that were used. The intent of the presentation is to present the background of the notation used for manipulating the score and to give a snapshot of typical applications of current computer music technology. The first part will discuss the representation of the score, and the conceptual tools that were used for structuring it. Part two steps through the realization process and the generation of processing and mixing scripts in a variety of formats for the composition. Every attempt has been made not to use advanced computer jargon without definition, although a general acquaintance with the techniques of contemporary music and the terminology of music in general is assumed.

## Background

The composition of *Kombination XI: A Ritual Place for Live and Processed Voices* began in 1978 with the recording of two voices reading a poem entitled *Kombination XI* by the contemporary German poet Helmut Heissenbüttel. The plan in 1978 was for a *musique concrète* realization at the electronic music studios of the Vienna Music Academy where I was then a student. The compositional approach I planned then was a relatively straightforward liturgical setting of the text using timbres derived primarily from one of the recorded voices. The goal of the compositional technique was to create a union on several levels between the text and its accompaniment—the

semantics of the text should be mirrored in the musical structures, and the prosody of the text should be mirrored in the melodic/rhythmical materials. This was my first electronic composition for voice, and I wanted to find a synthesis between poetry, speech semantics and musical form along the lines of baroque liturgical music.

The process of this realization was interrupted for twelve years by my moving from Vienna to Paris in the summer of 1978 to work at IRCAM and getting involved in computer music via software sound synthesis there. In late 1989, I decided to realize the *Kombination XI* using *musique concrète* techniques in software and using the unique facilities of the computers available to me for the two hardest processes of the realization: the management of the score, and the timbre processing in the preparation of the sound materials.

## The Composition

*Kombination XI* is the first movement of a four-part spiritual piece entitled *Celebration*. It is a rather dark, slow lament of intense grieving written for one or two live speakers (i.e., *recitative*), and recorded tape. The performance is intended to suggest a ritual or liturgical setting by having the speakers open with a recitation of the text (in the local language) over a pedal tone. Six episodes of approximately two minutes each follow the recitation on tape—a prelude, the four stanzas of the text, and a postlude—over the same pedal tone. During the tape part, the speakers perform a simple ritual involving several candles (the element of fire) on stage. The music of the episodes consist of over 2000 individual sounds derived from fragments of the voices of the two recorded speakers.

The male voice reads the text understandably, in sequence and without repetitions, using an intonation which was generated by phase vocoder resynthesis and soundfile splicing (described below) to re-interpret the original reading according to the compositional goal. The female voice is used more as a musical instrument for its timbral wealth. Musically we then have the elements of the pedal tone, the recitative, the male voice's reading of the poem, and the female voice's textures for the episodes.

The goal of the score representation was two-fold: to allow me to notate and manipulate the score's structures with a high degree of flexibility in order to achieve the desired "tightness" of materials; and to provide me with a notation that spanned

the expressive and structural realms with one set of notations and software tools. Because I planned to regenerate the entire text, I needed prosody editors with which I could edit the intonation of spoken text and resynthesize utterances to generate “my own” reading of the poem using somebody else’s voice. Because of the planned link between verbal semantics and musical structures, I needed tools with which I could manipulate middle- and higher-level musical structures. The idea of using a family of weighted binary trees based on prosodic stress trees and prolongational reduction trees came after I read Lerdahl and Jackendoof’s *A Generative Theory of Tonal Music* [2], which mentions this possibility in passing; it seemed like the ideal set of conceptual and notational tools for realizing the planned work. The use of concrete software tools based on these concepts and notations in the realization of *Kombination XI* will be described in detail below.

## The Score

### Tension-Relaxation Trees and the Score’s Representation

The primary structural abstraction used in the score of *Kombination XI* is a weighted binary tree based rather loosely on the prolongational reduction or tension-relaxation (T-R) tree described by Fred Lerdahl and Ray Jackendoof in their monumental book *A Generative Theory of Tonal Music*, and in the paper *A Computer Aid to Composition* [3]. 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).

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 three types of prolongational branching are:

replication or strong prolongation (denoted by an empty node in the graphical representation);  
less-stable repetition or weak prolongation (denoted by a filled-in node); and  
change or progression (denoted by no node).

These are illustrated in Figure 1, which shows the three types of tensing and relaxing patterns.

[Figure 1: T-R Tree node types]

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 *Kombination XI*.

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). Another paper [4] presents the implementation and use of the T-R tree system in the MODE software environment in much more detail.

## Examples of T-R Trees

Figure 2 shows examples of the different types of derived trees for several musical and uttered phrases. 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 a different musical structure.

[Figure 2: T-R and Prosodic Tree examples]

There are six central T-R trees from which much of the material and processing in *Kombination XI* is derived. They were, in turn, derived from six sentences of the text

of the poem. The first half of one of them (for the second-to-last sentence of the poem), is shown in the time-reduction and “sharpness” trees in Figure 2. Note the differences between these trees. The left-hand one illustrates the use of time in the utterance, whereby the “higher-level” branches go to the longer syllables. The right-hand tree is a weighted tree based on the subjective values of “sharpness” (roughly correlated to spectral brilliance), that were assigned to the syllables. Note here that the syllables with sibilant or fricative sounds are much “sharper” than the others.

## Score Management

### T-R Trees and Score Hierarchy

The abstract structure of the score for *Kombination XI* is expressed as a set of T-R trees which define a particular reading of the poem—an interpretation of the male voice’s part that is consistent with the T-R trees used for the rest of the piece. These trees are applied to various properties the sounds of the female voice in the episodes that accompany this reading. The top-level score consists of the main trees that define the reading (from which the male voice’s reading was synthesized), and the six sets of trees derived from these for the musical structures of the episodes (prelude, four stanzas, and postlude). The trees control the selection of source material, processing, layering and texture, and episode macro-structure. The lowest level of the score is six sound mix scripts using a list of about 2000 “particle” sounds. The particles are either syllables or single-phoneme segments of the voices of the speakers at specified values of length and sharpness (magnitudes similar to duration and pitch, described below).

To generate these from the source recordings, it was necessary that the words of both voices be segmented into phrases, words and particles (phonemes). The words and particles were ordered in terms of two subjective magnitudes—perceived sharpness and experiential length. These magnitude properties were used for sorting the sounds according to T-R trees in the reconstruction of the text and the textures for the episodes. The sharpness of a sound is defined as its subjective brilliance or brightness; the length is a subjective measure of its experiential length. The ordering was done by playing pairs of sounds and ranking the pairs. The system then assigned numerical values (between 0 and 1) for these magnitudes, flagging me when there

were impossible sets of relationships. Note that these properties need have no absolute units from the user's perspective; they are "order-only" magnitudes.

Given the grouping of the particles in the words word of a phrase, and the sharpness and loudness trees derived from the original utterance, one can now reconstitute the phrase selecting particles according to the original tree or any other one, substituting particles where appropriate for the stress and tempo patterns of the given tree.

## Score Editors and Representations

The score (expressed in terms of the particle sounds, T-R trees and sound mixes), was managed using the MODE (Musical Object Development Environment) [5], a library of software modules (object-oriented classes) written within the Smalltalk-80 Programming System. Several different graphical and textual formats were used for generating, managing and realizing the 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 [6]); and graphical tools for the derivation, editing and application of T-R trees.

Figure 3 shows a simple T-R tree editor of the type that was used for generating the vocal parts of the score. The upper part of the figure 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 piece.) 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 rhythm values based on the ordering of the four available ones (labeled "k1" through "k4").

[Figure 3: EventList View with T-R Tree]

[Figure 4: Location Editor]

The text, graphical, and gesture input formats included MIDI keyboard capture (in the T-R tree and event list editors), graphical editing of trees, event lists and event generators, and procedural Smalltalk-80 programs. The output formats that were used for the segmentation, processing and re-mixing included cmusic (a Music-V-like sound compiler program for software sound synthesis) soundfile segmentation and mixing programs, phase vocoder processing scripts, MODE sound processing scripts, MIDI playback (for pre-auditing trees), and MacMix (Dyaxis) mix scripts.

## The Realization Process

To make the score's application more tangible, I will now describe the realization process of *Kombination XI*. The final phase of the realization of *Kombination XI* took place between November, 1989 and April, 1990 at the Center for Computer Research in Music and Acoustics (CCRMA) at Stanford University, the offices of ParcPlace Systems, Inc., QuickSilver Studios, and the author's home. The configuration of the computer networks is shown in Figure 5 below. It shows the two Ethernet local-area networks at CCRMA and ParcPlace Systems connected via the Internet wide-area network. The Sun Microsystems SPARCstation computer in my home studio could communicate (via a high speed modem) with ParcPlace Systems (and from there with CCRMA) for transferring score and sound files.

[Figure 5: The Networks at ParcPlace Systems, CCRMA]

## Segmentation, Analysis and Mixing

The first step was the analog-to-digital conversion of the analog tapes of the two voices recorded in Vienna in 1978. An Ariel model DM-N digital microphone was used with an analog tape recorder connected to its line inputs. This was captured using the NeXT workstation's sound recording program and segmented into phrases (and in some cases words), using the NeXT system's excellent tools for soundfile editing including special versions of the SoundEd program by Lee Boynton of NeXT, Inc. and James Pritchett of Princeton University [7]. The user interface of this tool is shown in Figure 6, which shows a sound editor window, a spectrum view and the program's main menus.

[Figure 6: NeXT Soundfile tools]

These soundfiles were moved (on NeXT optical disks) to ParcPlace Systems for the long and quite compute-intensive process of segmentation into particles; analysis, classification and ordering; phase vocoding; and mixing of the reconstructed phrases and textures.

The MODE package was extended for *Kombination XI* by the addition of software for sampled sound and sound file processing and editing. The “Sound” class in MODE allows for multiple reference points (cues) within a file, along with arbitrary meta-information—e.g., what the utterance is, or what processing script was used to create the sound. Sound view and inspector tools were created to provide simple accessing methods to sound properties and help with the management of large collections of sounds. An example of a sound view is shown in Figure 7; it illustrates both the sound editor view and an object inspector on the sound being edited.

[Figure 7: Sampled Sound View]

## Particle Processing

The phase vocoder [8] is a software tool that performs digital frequency/time analysis and resynthesis of a sound signal—i.e., it takes an input sound, analyzes it into a set of overtones expressed in terms of their loudness and pitch functions, and regenerates the sound based on this analysis data. It can provide many interesting effects in that the analysis data—the frequency/time surface of the signal—can be modified between the analysis and resynthesis steps. Vcoders of various types have been used to great effect in voice-based computer music, especially by the masters Paul Lansky, Charles Dodge and J. A. Moorer. One can, for example, change the pitch and/or time contours of the sound independent of one another—e.g., to make it longer in duration without getting lower in pitch. In the extreme case (used quite heavily in *Kombination XI*, and the rest of *Celebration*), one can radically alter the pitch-time contour of a sound—e.g., to change the intonation of a spoken utterance or the melody of a musical phrase. In concrete terms, the phase vocoder used here was a software program written by F. Richard Moore at U. C. San Diego (described in detail in his book [8]), running on a SPARCstation computer, which is started with a command

that provides it with a number of parameters describing the desired processing of the input sound.

## **Managing Processed Particles**

The phase vocoder processing was driven by the score's list of required particles—the 2000 lowest-level sounds used in the mix scripts; because this process is quite compute-intensive, it was undertaken on the (very fast) SPARCstations at ParcPlace Systems. The sound file mixes for the reconstruction of the male reading, and for the female voice's episodes, were realized using the three formats mentioned above: MODE sound processing scripts, cmusic sound file instrument note lists, and MacMix mix files. Many of the larger rhythmical patterns were stored from MODE as cmusic note lists for an instrument library consisting of simple sound file input, scaling and panning instruments. These processes ran primarily on the NeXT network at CCRMA because of their optical disk storage whereby large numbers of sound files could be kept on-line during the execution of a mix.

The Studer/IMS Dyaxis digital mixing system at CCRMA was used for the episode mixes and the mix of the pedal tone. About 120 soundfiles were transferred from CCRMA's NeXT network via their Ethernet-to-AppleTalk network gateway to the Macintosh/Dyaxis system's local disks. The episode scores were written out from MODE in MacMix's simple ASCII mix format and were transferred from the SPARCstation at my home to CCRMA via the telephone and the Internet. This system was chosen because of the ease of editing the mixes using the Dyaxis system's MacMix software (written by Adrian Freed [9]). The results could be heard in the (excellent) listening room at CCRMA or written in digital format directly to digital audio tape (DAT) from the Dyaxis. Figure 8 shows an exemplary screen from the MacMix interactive user interface that runs on the Macintosh/Dyaxis system. It shows a number of sound files being mixed together.

[Figure 8: MacMix View]

## **Textures and Patterns into Episodes**

The score prescribes a 17-minute pedal tone consisting of a thick beat-frequency pattern of sine waves centered around 50 and 100 Hz. and a narrow noise stream (a whistle) centered around 1 kHz. which pans from right to left every two minutes (the

episode frequency). The music scores for this were realized at CCRMA and moved to the Dyaxis for the dry mix with the episodes.

The “final” mix and reverberation took place in April, 1990 at the QuickSilver (as in Messenger Service) Studio in Marin County, California. The studio has an exceptionally clean mixer for DAT-to-DAT mixes and reverberation fitting for *musique concrète* via a (good old fashioned) EMT echo plate (the primary echo device of the 1970’s) and a Quantec room simulator reverberation unit (borrowed from Loren Rush for the mix). The episodes and the pedal tone were played from separate DAT machines and mixed through the reverberators back to a third DAT recorder. Several mixes were produced of the final version, with different versions of the recitative—e.g., none (for performances with live speakers), German-only, or German and English (the normal version for tape performances).

## Conclusions, Future Directions

There are several novel aspects of the process of realizing *Kombination XI* that I hope will be attractive to other composers—perhaps even attractive enough to get equipment manufacturers to take notice. The first is the use of powerful high-level abstract notations for the score—with the requisite provision of text-based description languages (based on the Smalltalk-80 programming language) and graphical editing tools for generating and manipulating these structures. The fact that the T-R tree abstraction is very powerful and that it scales well from the lower-level expressive domain up to the very high structural levels is also worthy of note; this is one of the most wonderful aspects of Lerdahl and Jackendoof’s generative theory.

A separate issue is the possibility of having one score format and several command formats or script languages for the process of realization. The ability to use appropriate hardware and software for the various processes of the realization—rather than having to do everything with the same tool because of the lack of interoperability of tools—made the realization very much easier. This also facilitated the use of several types of computers, chosen because of their processing speed, storage facilities or special I/O interfaces. Another obvious ramification of this is the ability to use all available computational resources in the realization of a musical

idea. In the case of *Kombination XI*, the outcome would certainly have been different had it taken three or five times as long because I had been forced to use less-powerful computing systems.

In the future, of course, this should all be commonplace and mundane (he chuckles). I believe that it is however to be hoped that as more composers become more computer literate, music processing software and hardware systems will increase in their level of abstraction beyond the current (appalling) level of the note-editor and tape recorder paradigms that are ubiquitous in current MIDI and signal processing tools. The whole idea of computer-based software tools for composers (rather than for performers or copyists) is (sadly) still rather a novel idea; we can all hope that there will be more software for composition—as separate from notation or performance—in the future.

The real bottom line, however, remains musical, and I can say that I am happy with the musical results of the process described above (as is Mr. Heissenbüttel as well). I don't believe it would have been possible to realize this piece according to the original plan (as a tape piece), and I am currently using the same hardware/software systems for the realization of the further three movements of *Celebration*—two more rather dark laments (based on poems of the T'ang dynasty Chinese poet Tu Fu), and an ecstatic celebrative finale (based on *Simple Truths* by the contemporary American poet Albert Goldbarth). The abstract and concrete tools developed for this project are an extension of my previous work—a series of structural notations and graphical editing tools developed in various environments over the past eleven years and reported in the computer music literature. I believe that the abstraction of T-R trees and the software tools built to make it available to the working composer is an advancement of the art.

## References

- [1] H. Heissenbüttel, *Kombinationen* (Esslingen: Bechtle Verlag, 1960), also in W. Fehse, ed, *Deutsche Lyrik der Gegenwart* (Stuttgart: Reclam/Universal-Bibliothek, 1970).
- [2] F. Lerdahl and R. Jackendoof, *A Generative Theory of Tonal Music* (Cambridge: MIT Press, 1983).

- [3] F. Lerdahl and Y. Potard, *A Computer Aid to Composition* (Paris: Rapport de l'IRCAM, 1982).
- [4] S. T. Pope "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)", *Proceedings of the 1991 International Computer Music Conference* (San Francisco: Computer Music Association, 1991).
- [5] S. T. Pope "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, 1991).
- [6] S. T. Pope, "Modeling Musical Structures as EventGenerators", *Proceedings of the 1989 International Computer Music Conference* (San Francisco: Computer Music Association, 1989).
- [7] D. Jaffe and L. Boynton, "An Overview of the Sound and Music Kits for the NeXT Computer", *Computer Music Journal* 13(2): 48-55 (Summer, 1989) also in S. T. Pope, ed, *The Well-Tempered Object: Musical Applications of Object-Oriented Software Technology* (Cambridge: MIT Press, 1991).
- [8] F. R. Moore, *Elements of Computer Music* (New York: Prentice-Hall, 1990).
- [9] A. Freed, "MacMix: Mixing Music with a Mouse", *Proceedings of the 1986 International Computer Music Conference* (San Francisco: Computer Music Association, 1986)

## Figure Captions

Figure 1: The first examples demonstrate the types of branching described in Lerdahl and Jackendoof's generative theory of tonal music, as well as several prosodic stress trees for several common English words. Figure 1 shows the graphical representation of the six types of branching defined by the generative theory.

Figure 2: Figure 2 shows several musical and textual examples of the different domains of the theory. Figure 2a presents four examples taken from [2]: a time-span reduction of the Bach chorale "O Haupt voll Blut und Seele;" a time-span reduction of W. A. Mozart's *D Major Quintet* (with the metrical grouping shown below the staves); a prolongational reduction of the same (in its most condensed form); and a prolongational reduction of the *C Major Prelude* from *The Well-Tempered Clavichord* by J. S. Bach (with the arpeggii reduced to chords and the implication/subsumption of chords denoted by the brackets under the staves). Figure 2b shows the prosodic stress trees for two common English words; the direct lines from the top of the tree go to the most-heavily stressed syllables. In Figure 2c, two trees are shown for part of the second-to-the-last sentence from the poem *Kombination XI* (which means "And the questions are the sentences"). The left-hand tree is a time-reduction tree where the duration of the various syllables is denoted by their "weight" within the tree, and the right-hand tree is a "sharpness" tree where the syllables are weighted by their subjective brightness of timbre.

Figure 3: One of the Smalltalk-80 event list views (score editors) from the author's MODE software system is shown here. With this view, one can edit and apply prosodic stress trees derived from the properties of recorded spoken utterances. The upper part of the view is the tree editor where the nodes of the tree may be moved using the computer's mouse. The result can be written out to a file in the form of a script for the signal processing programs used for speech processing with MODE. The lower part of the view shows the event list being edited, in this case the German word *dunkelkammergespräche* (meaning "dark-room-discussions").

Figure 4: This figure shows another MODE system event list view—in this case a location editor where left-right stereo space is mapped onto the vertical axis. In this view, a pair of sound streams that are panning back and forth are being edited in that their sounds are being selected from a specified set (named “k1” through “k4”) on the basis of the desired sharpness and length contours. The buttons along the left-hand side of the view are editor operations (scroll, play, select, group, delete, inspect, redraw, and clear); next to this panel is a slider for setting the loudness of the selected sound(s). To the right of the main sub-view is the palette of available sounds.

Figure 5: The electronic local-area and wide-area networks connecting the computers used in the realization of *Kombination XI* at ParcPlace Systems, Inc. in Mountain View, California, the CCRMA Center at Stanford University, and the author’s home studio are shown in this schematic diagram. Some of the special co-processors (e.g., the Dyaxis digital mixer), and input/output devices (e.g., digital-to-analog convertors or removable optical disks), that were used are also shown under the computers they are attached to.

Figure 6: The NeXT computer’s sound file editing tools written by Lee Boynton and James Pritchett are demonstrated here. In this figure, one can see the main sound view in the upper-right part of the figure. The sound is the German word *bins* (am), and a small portion of the “i” sound is selected, as indicated by the gray vertical region. To the left of this view the program’s menus are visible. At the middle-left of the view is a spectrum (in amplitude vs. pitch coordinates) of the selected portion of the sound; it indicates that the “i” sound is rather harmonic as evidenced by the peak shown at 208 Hz. Below this is a help window showing the beginning of the on-line documentation of the program. In the lower-right corner is a sonogram view (in pitch vs. time coordinates) showing the spectrum of the entire word; the noise of the “s” sound at the end of the word is clearly visible from this view.

Figure 7: A MODE system sampled sound view and inspector are shown here. The sound view includes has cue points (cursors) set, and the tool’s menu is also visible. The inspector view below this shows the basic properties of a sampled sound.

Figure 8: This figure shows the user interface to Adrian Freed's MacMix sound editing program that is the primary front-end to the Dyaxis digital mixer. In this window, the sound files are listed along the left-hand border and time runs from left to right. The numbers along the top of the window are the durations of the entire mix file, and of the selected portion. At the right are faders for changing the loudness of the sound files. This example is a section of the score the for the first movement of *Kombination XI*.

# Figures

Figure 1

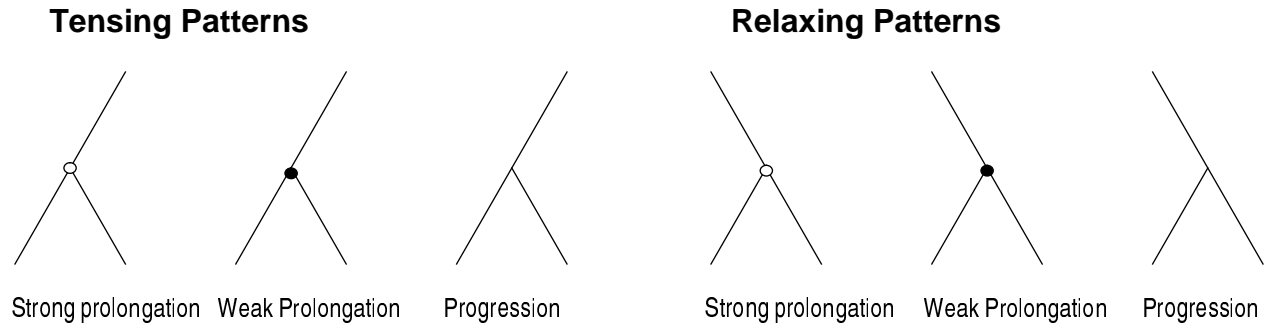


Figure 2a

## Melody Examples (from Lerdahl and Jackendoof 1983)

- Fig 5.8--p. 115
- Fig. 10.8--p.259
- Fig. 10.9--p. 260
- Fig. 10.11--p. 263 (top half)

(Figures 2a are from *A Generative Theory of Tonal Music* by Fred Lerdahl and Ray Jackendoof, MIT Press, 1983, used by permission)

Figure 2b

## Prosodic Stress of two Words

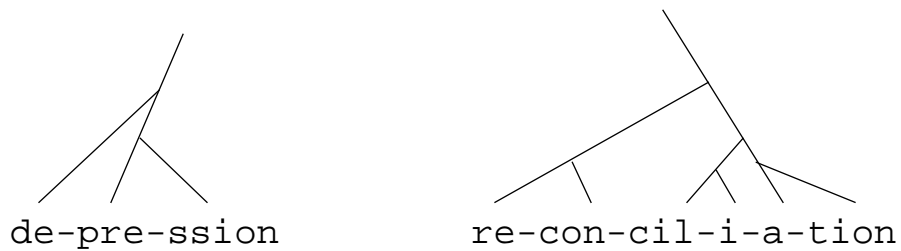


Figure 2c

**Time Reduction and “Sharpness” trees for a sentence from *Kombination XI***

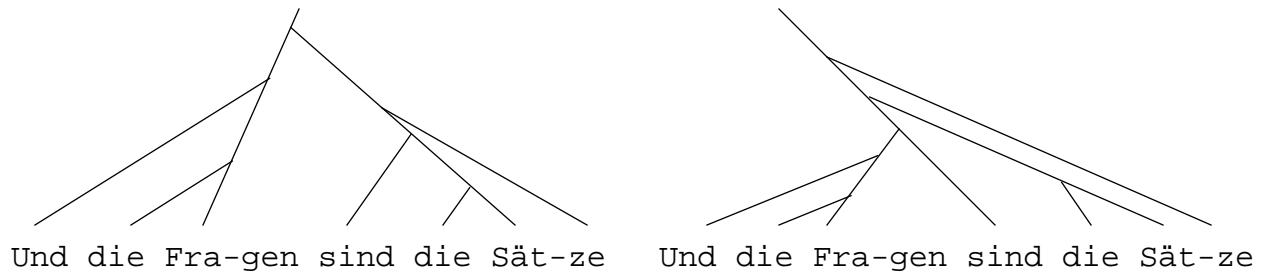


Figure 3

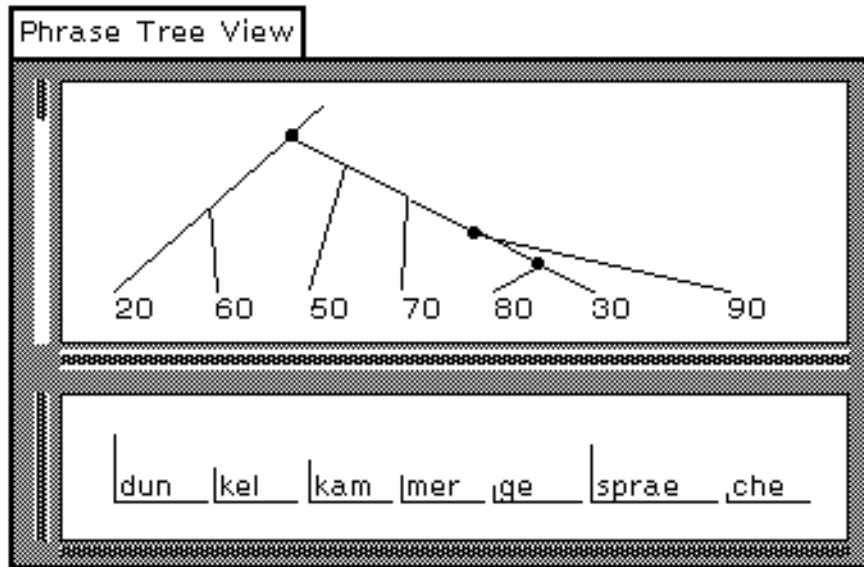


Figure 4

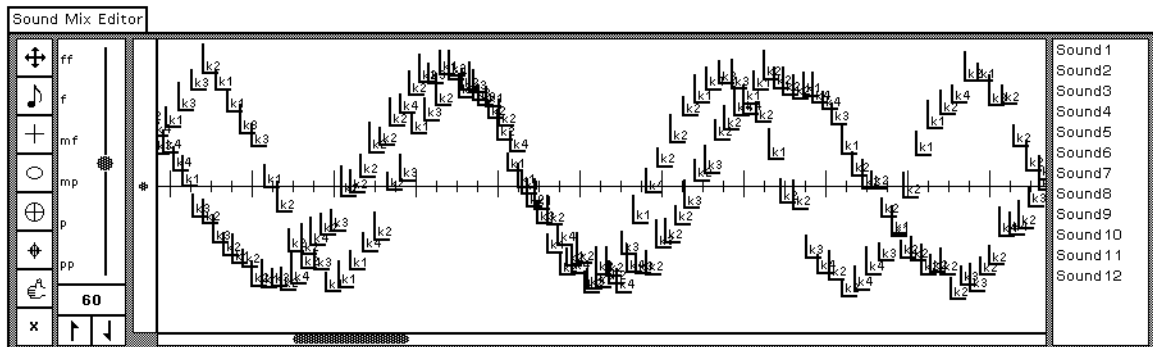


Figure 5

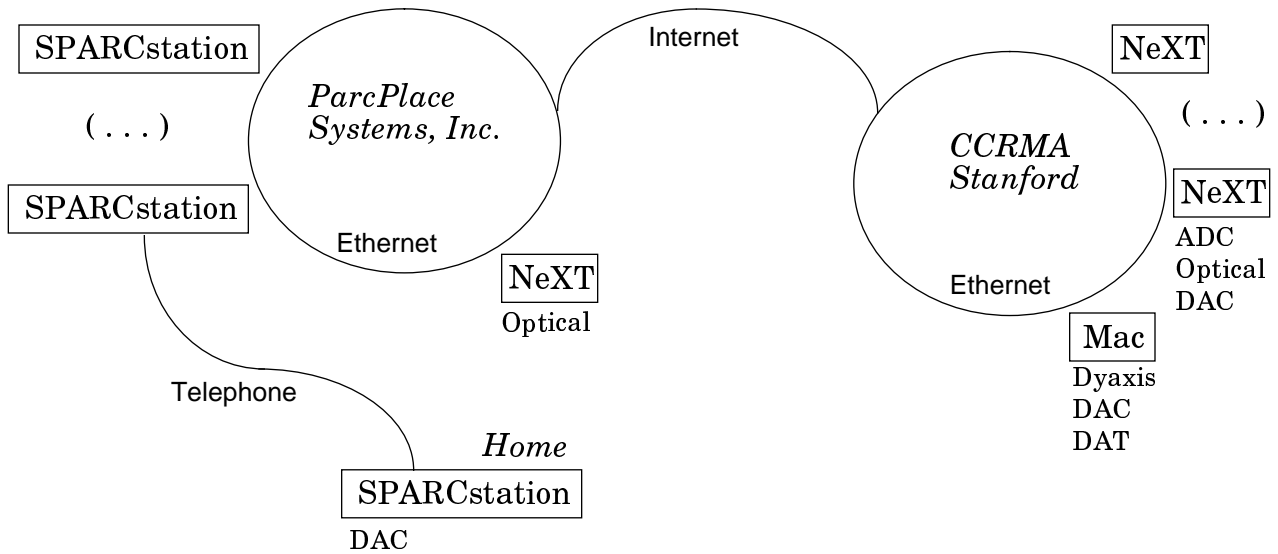


Figure 6—(SoundEd picture)

Figure 7

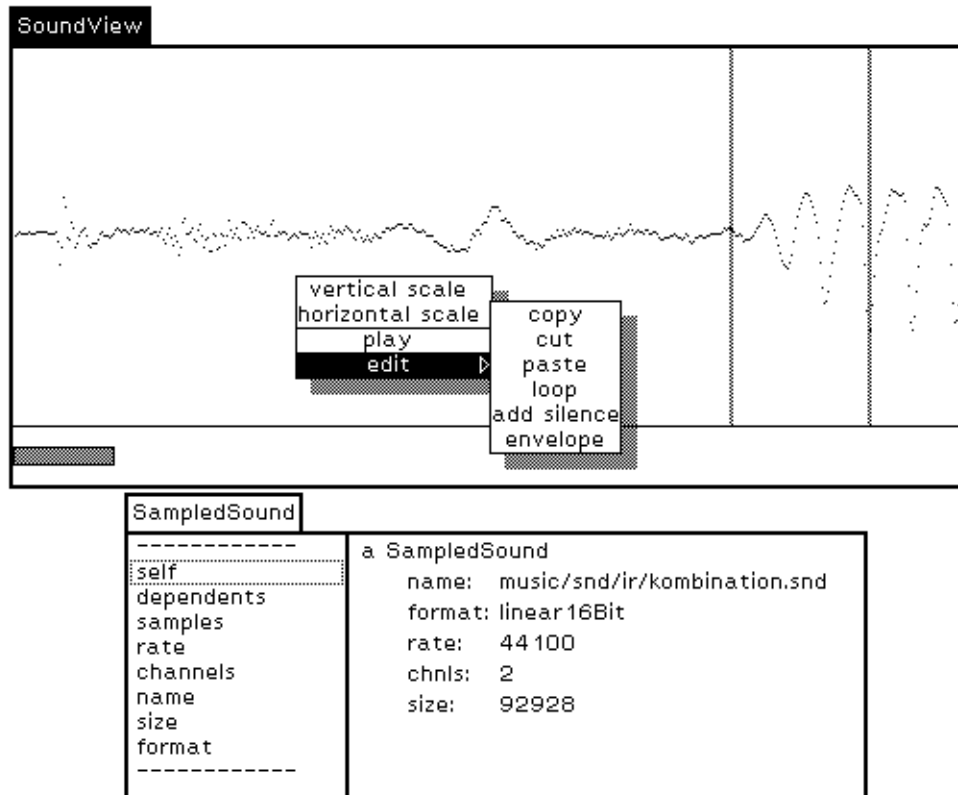


Figure 8—(MacMix Screen dump)