Composer's Underscoring Environment (CUE)

David Cope

University of California, Santa Cruz
Santa Cruz, CA 95064
EMail: howell@cats.ucsc.edu

Abstract

This paper introduces the Composer's Underscoring Environment (CUE) developed as a generalized extension of the EMI (Experiments in Musical Intelligence) program previously reported at the 1987 and 1990 ICMCs (Cope 1987, 1990). This new interactive form of EMI differs significantly from the original subset in terms of CUE's ability to track a composition-in-progress and meld found harmonic, motivic, and structural elements with stylistic signatures found in a user's previous works. Written in Common LISP and using the Common LISP Object System (CLOS), CUE offers a support base for composers by providing various analysis tools and stylistically-sensitive algorithmic composition.

1. Introduction

1.1 Need for CUE

Most interactive computer composition programs generate new musical ideas from various mathematic, connectionist, or pattern processing algorithms (Rowe 1993). These ideas can be used or discarded as users see fit. Important as these techniques are to composers, however, they do not offer all of the potential that computational means can provide. Many composers seek algorithmic composition in a style akin to their own. There is, then, a need for an interactive program which can track a composition-in-progress and combine found harmonic, motivic, and structural elements with more generic elements of a composer's style to create situationally relevant music.

1.2 Background: EMI

The EMI—Experiments in Musical Intelligence—program, described in numerous sources (see Cope 1990, 1991, 1992, 1996), has composed examples of music in styles as varied as Bach, Mozart, Chopin, Joplin, and Gershwin. EMI uses analysis of music in databases, pattern matching for signatures, and recombination techniques as basic tools for composition. EMI, however, creates complete works rather than composing interactively with users.

1.3 Uses of CUE

The Composer's Underscoring Environment (CUE) uses a subset of EMI to analyze and replicate generalized style. CUE uses a separate program to hierarchically analyze a current composition, interactively explore solutions to certain problems, and produce alternative routes at musical pivot points. Furthermore, CUE works interactively with composers to experimentally extend passages, develop potentials of composer-created germ ideas, and offer new ideas when inspiration temporarily

wanes. CUE composes as much relevant music as desired—from a single note to an entire work—in a composer's general style as evident in the previously-composed music in its database and in the style of the composition currently being composed.

2. Overview

2.1 Algorithm

Figure 1 shows an overview of the CUE program. Initially composers must establish a database—previously composed works in standard MIDI format—and set the variables of the CUE pattern matching program. Composers may initiate the algorithmic composing program at any time thereafter. Music is input with a standard notation program and viewed through a series of view levels. The analysis program informs composers of the signatures of their style and the hierarchical structure of the current composition.

3. Algorithmic Composing Process

3.1 Rules inheritance

The CUE composing algorithm operates, in part, by replacing the destination notes of a given segment of music with equivalent notes found in another segment elsewhere in the same database. This new segment of music may subsequently move in different directions creating new combinations of music. The new coupling follows the local rules of connectivity of the originating work regardless of its style. This process, called *rules inheritance*, is capable of producing highly original recombinations especially when segments of notes are quite small.

3.2 An example

Figure 2a presents a Bach chorale phrase. Figure 2b shows the first chord of figure 2a now attached to a chord of another Bach chorale with exactly

the same notes as the second chord of figure 2a, but which progresses to a third chord which is different from the original. This process continues producing a completely different chorale phrase which nonetheless follows the local contrapuntal rules of Bach. The program,

however, possesses no such rules and is therefore capable of composing rule-genuine music in a wide variety of different styles.

Figure 2a) 127 mm 1-2

3.3 Signatures

CUE protects musical signatures—patterns derived from pattern matching according to processes described in Cope 1991 and 1996—from being broken into smaller segments for recombination. Such protection ensures that stylistic signatures will survive the recombination process.

3.4 Transformational processes

The algorithmic composing program also recomposes music by voice. This transformational process exchanges voices between like-functioned segments in much the same way that rules-inheritance exchanges entire segments (i.e., using equivalent first and last notes).

4. The Structural Analysis Program

4.1 Structural inheritance

The music resulting from rules inheritance generally wanders, however, without any sense of balance or development. Structural integrity requires a more metalevel formal logic. When called upon for algorithmic composition, CUE employs structural inheritance of the music in its database to achieve this logic. This inheritance requires a deep hierarchical analysis as well as a specialized search for structural patterns.

4.2 Hierarchical analysis and earmarks

CUE utilizes a special program capable of recognizing pattern similarities at a variety of levels. This analysis algorithm operates independently and quite differently from CUE's signature program. This hierarchical program recognizes patterns called *earmarks* along with larger structural templates. Earmarks play a critical role in CUE's ability to analyze and generate logical musical structures. Like signatures, earmarks are protected from recombination. Earmarks are also location sensitive so that they reappear in new compositions in logical temporal locations.

4.3 Role of earmarks

Earmarks indicate, at least to the experienced listener, important attributes about a listener's temporal position in a work and foreshadow particularly important structural events. Earmarks can even contribute to expectations of when a movement or work should climax or end. Studies of musical semiotics have revealed a great number of such gestures in music (Agawu 1991, Gherdingen 1988). CUE integrates earmarks seamlessly into their immediate environment.

5. Interface

5.1 View Levels

CUE offers a series of view levels in which to enter compositions. The first level provides a standard notation window. Each of the subsequent levels can be produced by clicking anywhere on the current view with succeeding views presenting more area of the score in the

vicinity of the clicked on music. While each level offers less opportunity for actual composition—entering notes becomes impossible since location is too imprecise—these deeper levels provide opportunities for the user to visualize larger frames of the in-progress composition than the restricted views that small groups of measures impose.

5.2 Hierarchical interface

The highest level, or the one showing an entire work, also produces a structural analysis of the composition. Figure 3 shows such a toplevel version of *Mnemonics*, a work composed interactively by the author and CUE. Signatures are shown here with ovals and an accompanying "S." The rectangle with accompanying "E" indicates an earmark. As previously noted, earmarks play an important role in music and portend critical junctures in a work's structure such as ends of sections, ends of works, and so forth. In this example, the "E" rectangle signals the end of the work.

5.3 Foreground

The lines and brackets above the music in Figure 3 represent the structural layers of *Mnemonics*. The layer nearest the music tends to reveal repetitions or near repetitions of individual phrases. CUE discovers such repetitions (the foreground) and brackets them for easy recognition.

5.4 Middleground

Middleground analysis shows the patterning resulting from groupings of like material intermixed with groupings of contrasting material. Such groupings may form non-linearly; that is, they need not appear contiguously or even in the same section of music.

5.5 Background

Background analysis incorporates complex structures of more than one idea, helping to form more substantial interpolations of contrast, repetition, and variation (shown just above the previously discussed middleground).

5.6 Algorithmic hierarchy

When individual or grouped sections are repeated or varied at larger structural levels in music in a database, such repetitions or near repetitions are formed during algorithmic composition at similar junctures. This higher level patterning creates more complex forms and is responsible for investing the work with a reasonably logical audible structure. Hence, algorithmic

composition will not just extend material, but can provide logical contrasts when appropriate.

Figure 3

6. Output and limitations

6.1 Output

Completed compositions may be printed from the notation view or output using standard MIDI files which can be converted to music notation and/or performance files using commercially-available applications. These compositions may also be performed using a MIDI interface at any time while composing.

6.2 Limitations

CUE is designed to imitate, not to create anew. Hence, highly original work must still be accomplished by users. CUE can, however, free composers from certain obligatory demands of note-to-note composition in order to make higher-level decisions.

6.3 Program availability

For those interested in using CUE for their own compositional work, a functioning version of the program, including all of the analytical, pattern matching, composition, and notation algorithms described in this article will be made available in 1997 either on the internet or as an adjunct to a new book titled *A Composer's Workstation*.

Bibliography

- [Agawu, 1991] Agawu, V. Kofi. *Playing with Signs*. Princeton: Princeton University Press, 1991.
- [Cope, 1987] Cope, David. "Experiments in Music Intelligence." In *Proceedings of the International Computer Music Conference*. San Francisco: Computer Music Association, 1987.
- [Cope, 1990] ———. "Pattern Matching as an Engine for the Simulation of Musical Style." In *Proceedings of the International Computer Music Conference*. San Francisco: Computer Music Association, 1990.
- [Cope, 1991] ———. *Computers and Musical Style*. Madison, Wisc.: A-R Editions, 1991.
- [Cope, 1992] ———. "Computer Modeling of Musical Intelligence in EMI." *Computer Music Journal* 16, no. 2 (summer): 69–83, 1992.
- [Cope, 1996] ———. *Experiments in Musical Intelligence*. Madison, Wisc.: A-R Editions, 1996.
- [Gherdingen, 1988] Gherdingen, Robert. *A Classic Turn of Phrase*. Philadelphia: University of Pennsylvania Press, 1988.
- [Rowe, 1993] Rowe, Robert. *Interactive Music Systems: Machine Listening and Composing*. Cambridge: MIT Press, 1993.