CONTEMPORARY MUSIC ANALYSIS PACKAGE (CMAP) FOR MACINTOSH

Peier Castine
Technical University Berlin
Strasse des 17. Juni 135
D-1000 Berlin 12
Germany

Alexander R. Brittinman
Eastman School of Music
26 Gibbs St.
Rochester, NY 14604
U.S.A.

Craig H. Harris
MusicAT
788 Anthony St.
San Francisco, CA 94103
U.S.A.

ABSTRACT: The "Contemporary Music Analysis Package" (CMAP) is a comprehensive set of computer-based tools for the composition and analysis of twentieth century atonal music, implementing techniques developed and documented by Milton Babbitt, Allen Forte, and others. CMAP was originally developed for the UNIX operating system, incorporating a series of UNIX filters into the package to allow for a flexible flow of information between the individual components of the entire software system. A new version of CMAP for the Apple Macintosh, which will be processed here, is now under development. The challenge involved in developing a Macintosh version was to provide similar flexibility in an environment which neither supports the UNIX concept of pipes and filters nor provides facilities for shell script programming.

CMAP on the Macintosh

The "Contemporary Music Analysis Package" (CMAP) is a computer-based tool for the composition and analysis of twentieth century atonal music. Originally developed for UNIX, CMAP provides a comprehensive set of set-theoretic and serial analysis tools. Modular design and an executive use of UNIX pipes provide a particularly flexible software design. A version for MS-DOS is also available, using an identical architecture. In response to the wishes of many musicians, a version of CMAP for the Apple Macintosh is now under development. Our design goal is to provide a system which conforms to the spirit of the Macintosh user interface while retaining the flexibility of the original CMAP package.

Most functions of CMAP on the Macintosh are available through an "Analysis Scratchpad." This is a window in which pitch class combinations can be entered, analyzed, and transformed. Set information is displayed in "analysis lines," which show the set class name, prime form, interval vector, invariance vector, and other related information for the pitch classes entered.

There are a number of ways to enter data into the Analysis Scratchpad. The user can type in pitch classes, using hexadecimal notation, decimal integers, or pitch class names; MIDI keyboards will also be supported. The notation used during data entry may be converted to another format if the user so wishes. There is also a command which can generate sets according to a variety of parameters.

Commands are also provided to test analysis lines selected by the user for properties of invariance and interval content; to perform set transformations such as complements, union, intersection, or difference; and to perform two-tone transformations (transposition, inversion, multiplication operators). The analysis lines generated by these commands are automatically selected in order to facilitate chaining of commands. This also allows a series of commands to be saved by any of the journaling programs available for the Macintosh, such as Macromaker or QuickKeys, and to be invoked with different analysis lines as starting parameters.

Facilities are provided for the user to append comments to the analyzed sets. This can be an aid in referencing sets in a musical work being analyzed ("This is bar 20 of Klavierstück IX"). Certain commands append comments to analysis lines (for instance, "Count Pitch Classes" appends a table of the frequency with which each pitch class appears in a tone row).

ICMC GLASGOW 1988 PROCEEDINGS
Auxiliary Windows

Additional functions are provided by auxiliary windows. The Row Matrix window generates pitch class and pitch order matrices for arbitrary rows entered by the user; it can also search for the occurrence of specific sets or for invariant transpositions in the matrix. The Set Complex window generates tables of K and Kb relations for a complex of set classes as defined in [Forte 1984] and [Forte 1973]. The Set Mapping window is used to show operations under which a set maps into itself, its complement, or into an arbitrary superclass. Finally, the Rotation window allows the analysis of vertical structures created by segmenting and overlapping rows; this is primarily used to create rotational array matrices as described in [Rogers 1968].

Implementation

In order to provide a fast and efficient implementation of the various software tools, a data base of set class information is stored in a form sufficiently compact to remain resident in memory. This data base is called the Set Class Table. All sets are stored as bitmaps; this limits memory usage to a minimum and allows for the implementation of particularly efficient algorithms for most operations. All analysis operations use fast look-up techniques to access data from the Set Class Table. These features are described in more detail in [Harris and Brinkman 1986], however, a number of optimizations have since been implemented.

Combining Commands

One of the challenges involved with porting a package of UNIX programs to the Macintosh, aside from reworking the user interface, is in providing a mechanism for the individual components to communicate with each other.

In the UNIX version of CMAP, each command is implemented as an individual program. This has the disadvantage that the Set Class Table must be loaded into memory every time a new program is started. This disadvantage is outweighed by the advantage that the user can arbitrarily combine different programs with each other in a shell script and pipe the output of one program to the next.

The current version of the Macintosh system software (6.0.5) does not provide any facilities for communication between programs, nor is there a facility for shell scripts.

---

The Analysis Scratchpad Window

---

Set Operations
- Generate Set Classes...
- Test Interval Vector...
- Test for Invariance...

Transpose
Twelve Tone Operation...
Complement

Intersection
Union
Difference
Symmetric Difference

Count Pitch Classes
Imbricated Subset
Find Partitions...

Comment available for manipulating Sets

ICMC GLASGOW 1990 PROCEEDINGS
There are, however, a number of journalling utilities available for recording user events such as mouse clicks and keyboard input. Combining one of these utilities with the Macintosh Clipboard and the generic editing commands Cut, Copy, and Paste would have made it possible to simulate UNIX-style piping of standard I/O. A more elegant solution was found by combining the various programs found in CMAP into one application. Instead of using the Clipboard, we assumed that the information generated by each command is automatically selected. Since each command acts on the information currently selected, we have a simple implementation of I/O redirection as found in UNIX; in order to pipe the output of one command to the input of the next, the user simply needs to chain the selection of menu commands, as is done by the Macintosh journalling utilities.

Conclusions

Memory-resident storage of the set class data base using bitmaps to store set data provides a means for an efficient implementation of analysis techniques. The Macintosh version unifies the CMAP package to a single program, again increasing speed while simultaneously providing a mechanism for combining commands in arbitrary ways and thereby furnishing maximal flexibility for the user.

References

[Babbitt 1961]

[Forte 1964]

[Forte 1973]

[Harris and Brinkman 1986]

[Morris 1987]

[Rogers 1968]

ICMC GLASGOW '90 PROCEEDINGS