A COMPUTER ASSISTED APPROACH 
TO MUSIC ANALYSIS

Bruce B. Campbell
School of Music
Michigan State University
East Lansing, Michigan 48824-1043 USA
(517) 353-5340

ABSTRACT: This paper describes an approach to the presentation of musical
analysis using a graphically-oriented computer system (Macintosh SE) and a
CD-ROM player. Music unfolds linearly in time, but a group of six structure
demands "paralleled processing" from the listener that is anything but linear. Random
access to an audio Compact Disk permits the construction of an exploratory, yet
firmly guided learning environment in which outlines, diagrams and commentaries
are linked to actual passages of music. The student is able to re-examine analytical
relationships independently and repeatedly until they are aurally apperceived.

THE PROBLEM

The traditional lecture format used for the presentation of musical analysis relies
very heavily upon the printed score. It assumes that familiarity with the score will
permit the student to recall the sound of a passage that is chosen for discussion.
Naturally, score-reading becomes more difficult as the number of staves, clefs
and types of instruments increase, and because of this, many anthologies for
musical analysis and consequently, many analysis courses, draw most of their
examples from the solo piano literature. Musical analysis generally has had to
rely upon visual examples to suggest ideas and concepts, which are, at base,
aural. This paradox presents a substantial impediment to many musicians.
Compositions for which no published score exists (often the case with
electronically-generated music) tend to be absent from courses in music analysis.

The use of musical examples which exist only in score format has tended to con-
strict the kind of analysis that is done. Large-scale connections and the perception
of non-contiguous relationships may be important goals of sophisticated analysis,
but it is difficult for the student to review them independently. As a result, the
student is often forced to accept the more archetypic conclusions of the instruc-
tor with scant aural confirmation, and the process of analysis is less a process of
active participation than passive acceptance. This becomes apparent when the
student is presented with another nearly contemporaneous work by the same
composer. What was learned from one example in sonata form, for instance, is
not always effectively carried over as a principle to the hearing of another.

Courses in score-reading and figured bass, while at one time an indispensable
component of serious musical study, are largely optional and attenuated at
present, at least in the United States. So if music analysis is to become a relevant

ICMC GLASGOW 1995 PROCEEDINGS
process of discovery, and a lifelong methodology for thoughtful musicianship, some additional ways to reinforce its observations is needed.

Recorded music is a universal expedient, but until recently, precise access to specific passages has been impossible. Exceptons on cassette tape may be assembled tediously in advance by an instructor and used in their pre-established order. For a formal expository lecture this works well, but in the workshop atmosphere that is so much a part of courses in musical analysis at all levels, this procedure dissuades involvement.

THE IMPLEMENTATION

Recent advances in CD-ROM technology and software allow access to an audio Compact Disk at the block level: 1/5 of a second. This possibility opens up a wide range of modes of presentation of traditional techniques of analysis, and makes feasible many new ones.

The demonstration program was written in HyperTalk, using Apple Computer's HyperCard and CD Audio Toolkit. There exists a somewhat disdainful attitude by some Pascal and C programmers toward HyperTalk, which needs to be addressed briefly. HyperCard is often seen as a sorely needed response to the intricacies of Macintosh programming. But more than that, the concept of "hypermedia" goes beyond the bounds of traditional procedural languages, and in many ways, has different objectives. In any fairly elaborate "stack," methods and data are closely interrelated in a pseudo-object-oriented environment (hence the need for a new name, "stack" to replace "file"), whereas successful Macintosh applications, beyond smallish utilities, have tended to be "tools" with which one creates or manipulates separate data files (MacWrite, MacPaint and a host of progeny). Such programs are a means to an end, just as musical instruments are "tools." If one has a thorough background in a procedural language along with a considerable amount of programming experience, HyperCard may be deployed as a very flexible, powerful, and at times elegant language, especially with the extensibility provided by external procedures and functions (XCMD's). For the purposes of instruction, its interpreted nature provides no obstacle besides occasional annoying slowness. HyperCard permits an instructor to achieve rapidly the realization of presentational goals, while preserving the fundamental characteristics of the user interface intact. Chief among these is that the user be always in control of the program. The design of the language makes it virtually impossible for an author to keep proprietary any methods or data. As a result, there has been little commercial interest in the product, but it is precisely the openness of the language that make it suited to the academic environment. Many instructors prefer to devise and distribute their own class materials, in preference to a static textbook. The stack may be distributed to students as a kind of "electronic handout," and just like their paper predecessors, can be subject to constant revision and refinement as the course progresses.

ICMC GLASGOW 1990 PROCEEDINGS
Several issues were seen as crucial in the design, raised by the observations of many years teaching analysis at the undergraduate and graduate levels. The crux of the matter may be summed up simply as "too few." Students hear too few interpretations of a piece (often, only one recording), and they hear too few examples of a formal archetype, so that they are really learning factsoids about one particular piece, rather than gaining a feel for structure as a musical dimension. The breakthrough to some kind of instantaneous (often called "intuitive") recognition is too rarely achieved.

Firstly, different performances would have to be accessible, so the student would learn the music, and its structure, apart from anyone's particular interpretational viewpoint. As an alternative to a great number of global variables, a scrolling field for each Compact Disk is set up, keyed to another field that contains comments and which often functions as the source of on-screen remarks or dialog. It is a straightforward process to add new performances as they become available. Secondly, the format would have to conform to some standard of presentation. This would permit the instructor to prepare readily quite a number of examples for each formal archetype, and would allow the student to use them with a minimum of readjustment for each piece.

Other factors seen as important were: a single-screen overview that would allow access to any of the major formal divisions, a detailed examination of the composer's down to the level of the phrase, and a self-test that would prepare the student for an in-class quiz. (The last is a practical matter, but based upon the belief that one really doesn't know a piece thoroughly until one can identify the context of any of its component parts.) A series of screens constitutes a table of themes: the student can hear any instance of a particular theme. There is an emphasis upon large-scale harmony as the foundation of formal articulation, and as the key to understanding the operation of transitional and developmental sections. On-screen buttons provide suggestions of what to listen for, and pose questions of a speculative nature, which encourage the students to forge for themselves those additional mental "hyperlinks" which are the basis of sensitive hearing.

SELECTED RESOURCES

Dartmouth XCMD's 3.1. Useful collection of external commands and functions for the academic environment, with source code for most given in Pascal or C.


Bond Gary. XCMD's For HyperCard. Portland: Management Information Source, Inc., 1988. Shows the intermediate-level programmer in Pascal or C how to create external code segments that can add features (or speed).