Demonstration of *Concerto Accompanist*, a Program for the Macintosh Computer

Frank Weinstock
Associate Professor of Piano
College-Conservatory of Music
University of Cincinnati
Cincinnati, OH 45221-0003, USA

Abstract

*Concerto Accompanist*, a program for the Macintosh, provides an intelligent accompanist for a pianist learning a standard-repertoire piano concerto. The pianist plays the solo part on a Disklavier (or any MIDI keyboard), and *Concerto Accompanist* controls one or more MIDI synthesizers playing the orchestral accompaniment, following the tempo and dynamic changes of the soloist. This is an update of the “music-minus-one” concept, the difference being that the orchestra follows the soloist, not the other way around. It is a practical program to help pianists practice, and as of yet does not have serious performance or research applications.

1. Description of the Program

At ICMC 1993, I will demonstrate *Concerto Accompanist*, a new program I have written for the Macintosh computer. By using the following metaphor for a piano-concerto orchestral, *Concerto Accompanist* provides an intelligent accompanist for a pianist learning a standard-repertoire piano concerto. The pianist plays the solo part on a MIDI keyboard (preferably an acoustic piano with a MIDI-out capability, such as the Yamaha Disklavier). The computer, running *Concerto Accompanist*, functions as the “conductor”, “listening” to the soloist and “conducting” the “orchestra” (one or more MIDI synthesizers) accordingly. The “conductor” must, of course, follow the “score”, which has been previously input into the computer and “marked” with hints as to what the “conductor” should “listen” for.

What differentiates *Concerto Accompanist* from the old-fashioned and relatively trivial “music-minus-one” concept is that it is able to respond to the way the soloist plays. Thus, not only is the soloist not limited to the tempo of the conductor and orchestra that made the “music-minus-one” recording, but also *Concerto Accompanist* reverts to changes of tempo and dynamics from the soloist—in a manner comparable to what a live conductor and orchestra would do. Thus the pianist is free to develop his or her own interpretation without saving to follow somebody else’s.

*Concerto Accompanist* runs on virtually any Macintosh (Plus or greater), and is fully compatible with System 7 (it requires System 6). It uses Apple’s MIDI Manager, and its memory requirements are small: it needs only 512 kilobytes, and takes up less than sixty kilobytes of disk space (the actual code is less than thirty kilobytes—the rest is taken up by such things as text for help balloons, etc.). The specially marked “scores” are a bit larger, with a first movement of a Mozart concerto taking up 150-170 kilobytes of disk space. Because of the need for maximum efficiency (the program needs to react very quickly to the soloist’s playing), the program was written in Assembly Language, using Apple’s *Macintosh Programmer’s Workshop*. The program is easy to use, fully complying with Apple’s user-interface guidelines; furthermore, the basic commands (play, fast forward, stop, etc., as well as setting several parameters) can be issued remotely from the MIDI keyboard (or piano), so that the pianist doesn’t have to run back and forth between the piano and the computer.

ICMC Proceedings 1993 461 7D.1
2. Applications

Concerto Accompanist is primarily a practical program, and does not pretend to yield serious insights into scientific areas of research such as artificial intelligence, which work has been pioneered elsewhere; nor does it have serious performance applications at this point. Its purpose is to be an aid to the serious piano student learning the concerto repertoire, replacing the pianist's need to impose on another pianist to play the reduction of the orchestra part; furthermore, synthesizers can sound much more like an orchestra than a second piano can, thus giving the pianist a better sense of what it is like to be in front of an orchestra. Since all that is needed is a computer, synthesizer, and software, rather than another piano and pianist, it is certainly a cost-effective alternative.

3. Considerations of Development

The score of a piece is in the form of a specially formatted file which contains all the orchestral parts and a "marked" version of the solo part. The marks provide Concerto Accompanist with hints as to how much importance to give to the way the soloist plays the various notes. For example, if the pianist plays the seventh note of a long sixteenth-note run earlier than expected, it is probably just the result of uneven finger-work and should be ignored; however, if the second quarter note of a melody at the beginning of a new section is earlier than expected, it likely indicates at least a temporary change of tempo, and the program should react accordingly. Thus, to a certain extent, the program's effectiveness depends on how well the score was marked. (The user is not expected to do the marking, but would purchase — once the program is commercially available — a previously marked score.)

In marking the scores, one of the interesting questions has been how to deal with the common situation, in which the soloist and orchestra are to play notes at the same time. A human conductor could certainly not wait until (s)he heard the soloist's notes and then signal for the orchestra to play — it would be too late; rather (s)he would anticipate the beat based on previous notes, and conduct accordingly. However, because of the speed of the computer/synthesizer's response time, it seems that the "wait and react" method often (but certainly not always) works better for Concerto Accompanist, particularly for major downbeats, etc. This is a good example of computers and humans accomplishing the same task in substantially different ways; it is perhaps analogous to a chess program evaluating positions by playing out all possibilities several moves into the future, rather than by using more "intelligent" and abstract criteria, as a human player would — brawn over brain.

The biggest challenge in writing the program has been to deal with the less-than-perfect — i.e., any — pianist. If we could count on the pianist to play every note at the right time, and not miss or add anything, the programming would be relatively trivial — the program would need to just monitor the soloist and adjust the tempo accordingly. However, interpreting and reacting to various mistakes on the part of the pianist poses a number of interesting questions and leads to certain trade-offs in the program's efficiency and effectiveness. The difficult thing is to understand what caused the mistake. For example, assume that the solo part includes a C-Major scale in fast notes; the computer hears the C, and then an E, about halfway between when it expects the D and E. Did the soloist miss (i.e. not play) the D and play the E early, or did (s)he try to play the D late, but hit the E instead? It's not always easy to tell.

4. Future Development

Future plans for the program, other than continuing to improve its current abilities, involve accommodating more possible applications — i.e., no longer limiting its use to the rehearsal of piano concerti. This includes making it compatible with any MIDI file, which would enable the user to select any part of a piece as the "solo" part, and have the synthesizer(s) play the other parts while (s)he plays the "solo" part. This would be a major task teaching Concerto Accompanist how to "mark" the solo part, a task that is now done by a human. Another goal is to incorporate pitch recognition into the program, so that the input could be from a microphone rather than a MIDI interface. This would take away the need for the "solo" instrument to have a MIDI-out capability — it could be a violinist practicing a string quartet, or a karaoke singer in a night club, for that matter. Less lofty goals include removing the dependency on Apple's MIDI Manager (which software has its advantages and disadvantages), as well as introducing versions for IBM compatibles.
5. Summary

Concerto Accompanist is now fully functional; it has been demonstrated on a number of platforms, and used by several people. As of this writing, it is not yet commercially available, but should be shortly. When it is, the whole system should be relatively inexpensive to install in piano studios everywhere, and would be a great help to the serious piano student and teacher. A pianist with a Diaklavier, Macintosh, and synthesizer could buy the software, and then buy the "scores" to individual concerti as needed. It would give pianists the opportunity to repeatedly hear and interact with the orchestra part of a concerto before facing the daunting experience of getting up to rehearse with a live orchestra for the first time.

6. Acknowledgments

The creation of Concerto Accompanist was supported in part by the Yamaha of America Corporation and the University of Cincinnati. The author is grateful to those two organizations, and to Drs. Robert Werner and Frank Tepe (University of Cincinnati), Dr. Joel Kabakov (Yamaha of America), Mr. Kenji Osako (Yamaha), and George Litterst (New England Conservatory of Music) for their interest and support.