Center for Research in Electronic Art Technology
University of California, Santa Barbara
http://www.create.ucsb.edu

Application & User Interface Development at CREATE

The history of computer applications in music reaches back into the 1950s. Only recently, however, has it been possible to control complex musical processes such as algorithmic composition or sophisticated sound synthesis programs in real-time. Advanced software and hardware technology also allow us to develop user interfaces that allow non-musicians (and even non-readers) to be musically creative. These two domains of application development and user interface construction have been important tasks at CREATE for ten years. We present examples of tools we've developed below, and discuss what features they introduce that might be useful to other application areas.

Background

In many domains of human creativity, computer technology has enabled new organizations of the common tasks so that "expert" users can be more effective and "novice" users can contribute to the field. In the area of writing, for example, we have sophisticated and flexible document preparation and layout programs that allow an author to write and produce a finished work to his/her own specifications, and also simple voice-input systems that allow non-typers (and even non-readers) to communicate via the printed page.

For computer applications in music, the historical development started in the 1950s with programming languages for music that were based on programming languages from other domains (e.g., FORTRAN). Later (1960s and onward), computers were used to control analog synthesizers. With the revolution in highly available personal computers and graphical users interfaces (GUIs) starting in the 1980s, a whole range of new applications in music appeared, but most of these stuck to tried and true operational paradigms such as music programming languages (possibly with GUI front-ends), and computer simulations of analog hardware (e.g., recording studio mixing consoles or tape recorders).

At CREATE, it has been our task since the early 1990s to move beyond these two categories, and to provide powerful and flexible tools to professional users, and very easy-to-use tools to non-experts. We are driven by the dual aims of enabling professionals to compose and produce music more easily, and allowing more of the general populace to be musically creative in their own rights.

This white paper describes several of our efforts in this area, and proposes several areas for future research and development.

Previous Work

Over the last ten years, CREATE developers have produced several well-known and widely distributed music software packages aimed either at professional composers and performers, or at non-experts and even complete neophytes. We will describe several of our results in the sections that follow. In each case, there are novel user interface techniques or innovative interaction methods that could find applications in areas other than sound/music production and signal processing.

The systems we present below are quite diverse; they are written in the C, C++, Smalltalk, Java, HTML, and/or SuperCollider languages, and are delivered on the Macintosh, UNIX/Linux, and/or PC platforms.

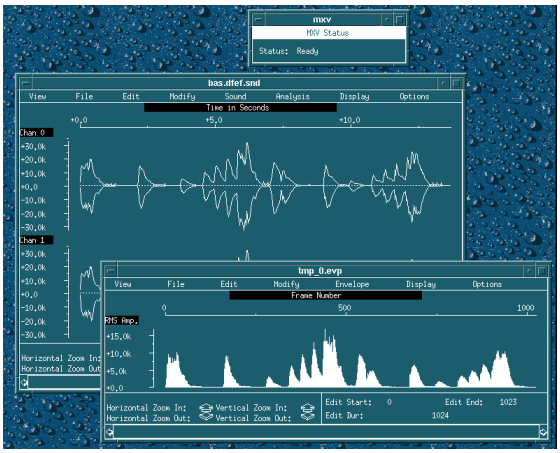
MiXViews

The *MiXViews* software package is a sound editor/processor for composers and producers; it runs on most flavors of UNIX/Linux computers. MiXViews allows users to record, edit, and process digital sound and analysis data derived from sounds.

A range of different kinds of data can be created, edited, and stored, all of which can be used in sound synthesis. Any number of files may be displayed simultaneously on the

screen, and one can cut and paste between them. Many operations on sound data are possible, such as time / amplitude envelopes, spectral filtering, and even full-blown analysis / resynthesis techniques such as the phase vocoder.

MiXViews has been distributed from CREATE since 1992, and is currently maintained and ported to new versions of UNIX and Linux; it has been used by most of the graduate composers that have passed through CREATE in the last decade.

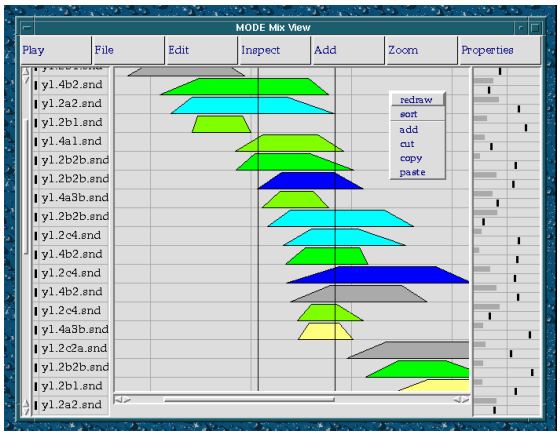


MiXViews time and envelope views

The MODE

The *Musical Object Development Environment* (MODE) and its successor *Siren* are extensible Smalltalk-80 frameworks and tool kits for music description, score editing, interactive performance, and digital signal processing. Both systems address five areas: (1) the representation of musical parameters, sounds, events and event lists; (2) the description of middle-level musical structures; (3) real-time MIDI and sound I / O and scheduling; (4) a high-level user interface framework and widget set; and (5) several built-in end-user applications for signal and event structure editing and digital signal processing.

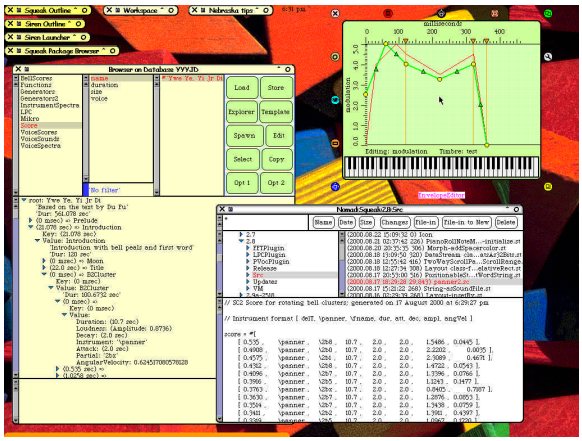
The MODE and Siren both support applications ranging from performance analysis to score image processing for musicology to interactive performance to large-scale sound / music databases. The MODE has GUI tools for sound mixing, analysis / resynthesis, spatial processing, MIDI sequencing, and music composition.



MODE mix view showing a time-line and several sounds to be mixed

Siren

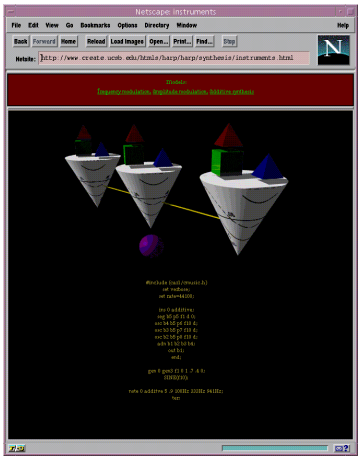
Siren is a new version of the MODE that is written in the open-source and very portable *Squeak* implementation of Smalltalk-80. Siren incorporates a flexible object database. Like MiXViews and the MODE, Siren has an active user community, web site and email list, is ported to new releases of its base software.



Siren GUI examples: score database, envelope function editor, and score file list

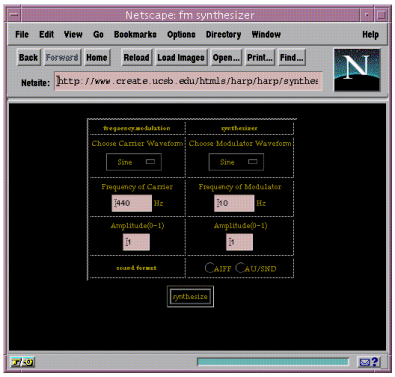
HARP

The potential of the world-side web for use in teaching is widely mentioned, and rarely exploited. The *HARP* web site was developed in the mid-1990s to teach sound design using client / server sound synthesis programs running on CREATE's servers, and interactive web-client-based GUIs that lead (possibly off-site) students through the steps of learning to use software sound synthesis methods.



HARP example for teaching additive synthesis over the Web

The HARP was very innovative for its time in the use of a web browser as a GUI to a server-based signal synthesis program and for streaming multimedia data over the Internet.

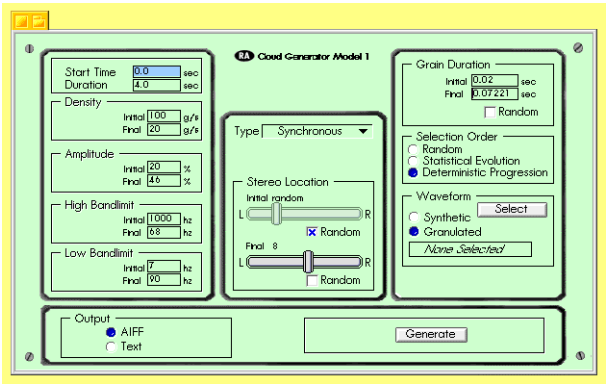


HARP example showing the editor dialog for learning FM synthesis

CloudGenerator

One of CREATE's other research areas is related to a family of new sound synthesis and processing techniques that are collectively known as granular synthesis. Just as light energy can be viewed both in terms of wavelike properties and in terms of particulate properties (photons), so can sound.

Microsound signal processing views all sound as a combination of elementary sound particles. The duration of these particles falls between the threshold of timbre perception (a few microseconds) and the duration of brief notes (a few hundred milliseconds). The micro-time scale has taken on increasing importance as a resource in audio engineering and music composition, because operations on this time scale cause dramatic acoustical effects on higher levels of sound structure.

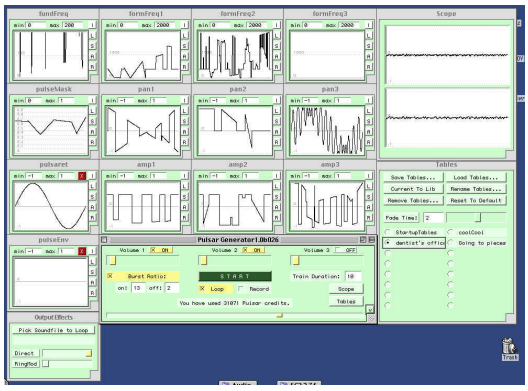


CloudGenerator configuration screen

The *CloudGenerator* program is a very simple GUI-based program for learning the characteristics of granular synthesis. It provides the user with a configuration dialog box (based on the look of a laboratory signal generator) in which the properties of a granular "cloud" can be set. The cloud can then be synthesized using a built in software synthesizer, or it can be stored as a score file for later off-line processing and / or synthesis.

PulsarGenerator

At the other end of the scale, so to speak, the *PulsarGenerator* program aims to provide a flexible and powerful interface to an extended (and very complex) granular sound synthesis and processing technique we call pulsar synthesis.



PulsarGenerator GUI showing table editors for various properties of algorithmic composition and synthesis

The PulsarGenerator GUI provides a user-configurable set of table editors that control both the compositional algorithms and the sound synthesis methods of the program. The internals are written in the SuperCollider programming languages, and are provided as open-source for user extension. The GUI is extremely flexible, and provides a number of powerful commands for real-time editing of control and signal tables. Among its innovations are:

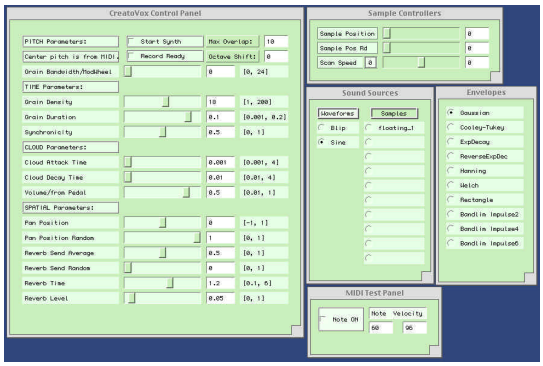
- import of sound files as control functions;
- copy, paste, mix of waveforms and envelopes, where the mix automatically scales to the range of the target function;
- loading and saving of wavetable banks
- real-time cross-fading between different wavetable settings;
- real-time wavetable editing and rescaling
- real-time output waveform scope; and
- real-time wavetable modifications.

The Creatovox Instrument

Our research agenda includes a real-time implementation of microsound synthesis and signal processing with flexible performance controllers: the Creatovox project. The hardware component of the Creatovox consists of a keyboard controller and a variety of continuous-range input devices such as sliders, pedals, 3D joysticks, etc., as shown in the figure below.

Curtis Roads performing on the Creatovox performance interface

The Creatovox GUI displays the state of a selected set of the input values, and provides basic control of the underlying SuperCollider synthesis engine.

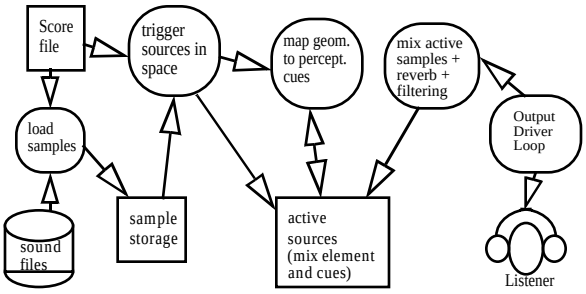


Creatovox GUI for monitoring parameters arriving from the physical instrument

The Creatovox is the first example of a planned series of hardware / software interfaces for controlling complex real-time software systems (see below).

The CREATE Auralizer

Another central focus of CREATE's R&D is surround sound and spatial audio production and performance. The *Creatophone* and *auralizer* projects are software and hardware systems, respectively, for spatial sound rendering and performance.



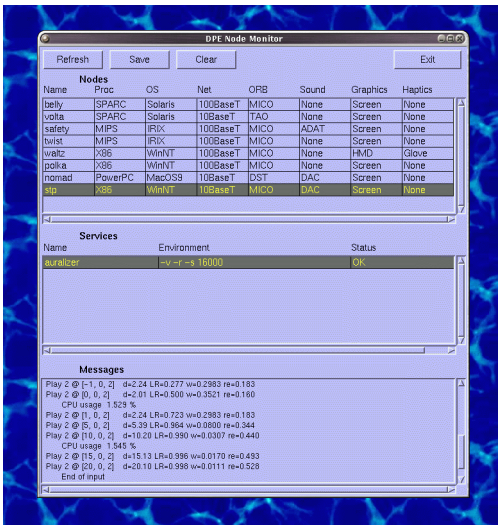
Auralizer system architecture

The auralizer is a flexible and scalable software system for taking sound (generated live or recorded) and processing it so that it can be made to appear to originate from any point in space. We have used this software in several of the "virtual reality" systems developed at CREATE. The auralizer is a scalable rendering engine that can produce from 1 to 32 channels of spatial sound cues.

The DPE Distributed Processing Environment

In order to manage large-scale complex programs such as the Creatovox or the auralizer, we need a management system for distributed real-time programs. The CREATE *distributed processing environment* (DPE) is a suite of programs (with associated databases and "interface repositories") for setting up, starting, and monitoring distributed real-time object-oriented software.

The DPE allows a user to configure how an application is to run on the various computers on the available network, and to start it and monitor the run-time performance of the system. We are planning to incorporate automatic load-balancing and fault-tolerance in a future version. The DPE monitor tool allows the "manager" to view the activity of various objects distributed on servers throughout a (local-area or wide-area) network.



DPE management GUI

The DPE tools can be used to configure and manage any large-scale distributed system where the components are stand-alone software services communicating over a network, e.g., for application servers, streaming content delivery, or media data processing.

Work on Immersive User Interfaces and Virtual Environments

The technology that used to be called "virtual reality" is now understood as immersive user interface technology, and it can be applied to a wide range of application areas. In some of these cases, the underlying application is a simulated "world" with which the user interacts, and in others, the user is exposed to a visual and audio environment that is derived from the features of some media object source such as scientific or artistic data. We are experimenting with the use of immersive user interfaces by creative artists (e.g., composers) and as interfaces to ease-of-use applications such as scripting systems for novice users.

Young user testing an immersive user interface with gestural input

Plans for Future Work

All of the systems described above fall into two categories:

- those that provide easy-to-use interfaces to allow untrained or novice users to perform multimedia creation or processing tasks, and
- those that present expert users with power and flexible user interfaces to perform extremely complex tasks efficiently.

We believe that these are two areas of application and interface design that have the potential of greatly increasing the effectiveness of software for digital media systems in the future.

As the survey of our work over the last decade illustrates, there are several areas of interface and application design that still need refinement for the next generation of media applications to be created. At CREATE, we intend to investigate each of the following topics in more detail:

- Ease-of-use systems for non-readers to perform scripting on virtual environments;
- Graphical editors for complex sound synthesis techniques;
- User interfaces for teaching sound design and synthesis techniques;
- Software support for spatial sound composition;
- Graphical browser-based user interfaces to multimedia databases;
- A sound / music recording / editing / publishing suite for non-readers;
- Management tools for large distributed systems;
- Computer vision and user tracking for real-world applications;
- Using spatialised sound for auditory data display.

The various spin-off product technologies from this R&D work include:

- new software and hardware support for sound and music production and processing;
- improved user interfaces for computer-based gaming and content;
- better immersive interfaces to simulation-based virtual environment applications;
- new tools for managing large distributed systems and application servers; and
- innovative interfaces to media databases and content retrieval and network streaming systems.

Contact Information

Stephen Travis Pope
Tel: (805) 967-2621
Email: stp@create.ucsb.edu

Updated 2001/3/15