Performing Music in three Dimensions
(Musikperformance in der dritten Dimension)

Johannes M Zmölnig*, Thomas Musil*, Winfried Ritsch*, Robert Höldrich*
* Institute of Electronic Music and Acoustics, University of Music and Dramatic Arts, Graz
{zmoelnig,musil,ritsch,hoeldrich}@iem.at

Abstract
Contemporary electro-acoustic music increasingly demands spatial positioning and movements of sounds.
During a performance, it is often necessary, to render both virtual sound objects (pre-recorded as well as live signals) and pre-recorded soundfields into variable virtual and/or "enhanced" acoustics.
To fulfill the manifold requirements of sound transformation and interactivity, a modular concept can provide the possibility for on-demand extensions.
Reproduction has to be possible on a variety of setups, ranging from binaural systems at the composer's studio to large scale loudspeaker arrays in public auditories.
In this paper, a periphonic software mixing system that aims to meet these requirements, is introduced and discussed.

1. Introduction
Since 1997 the IEM has been experimenting with periphonic production of electro-acoustic compositions, since 1999 the focus has been laid on ambisonic systems. Therefore, an ambisonic based mixing environment had been created to drive the IEM's laboratory and concert hall "IEM CUBE" [01].
This system had been designed to allow abstract movements of virtual sound objects in space, without the need for the composers to care about technical details like the loudspeaker setup. Additionally, this PC-based system was designed to be scalable, in order to be used both inside the laboratory (with powerful stationary hardware) and outside it (with mobile equipment).
Nevertheless it turned out, that the system was not accepted very well by composers and technicians: most existing electro-acoustic pieces would not use the abstract movement of sound objects at all, whereas newly commissioned compositions often had special requirements, that could not be accomplished by the general approach of the environment, which had little to no possibilities to adapt for new technologies or special needs. Only a handful of compositions where explicitly made for this system.
One of the major reasons to not accept the environemnt was, that it was unclear, whether the system should be considered a general mixing console or a "spatialization effect": on the one hand it did not provide the functionality of a traditional mixing console (grouping, submixes,...), on the other hand the hardware and skills requirements for mere spatialization were rather big.
2. Aims

In order to overcome the deficiencies of the existing periphonic environment, a new concept for such a system had to be established.

The main goal is to provide an environment to recreate a wide range of electro-acoustic pieces with focus on periphonic reproduction and virtual acoustics.

To allow intuitive handling of this system, the analogy of a traditional mixing desk was chosen: the system should be able to behave as an ordinary multichannel mixer, and at the same time provide new approaches to spatial composition.

A modular design should make the integration of new concepts and artistic ideas relatively easy.

The system ought to be scalable, in order to produce satisfying results in dedicated environments but also to be usable with mobile equipment in a variety of locations.

3. A periphonic mixer concept

In order to fulfill the requirement of modularity and extensibility, a software based approach is taken.

For performance reasons it is advisable to split the entire application into several modules, which can be used together in a monolithic application as well as run in separate threads, either on one powerful computer or on several machines.

Usually, an application is split into a DSP-part that runs with low latency at a high priority, and a GUI-part that controls the DSP [02].

More generally, tasks are separated according to their interaction with the "real world":

- a DSP module acts on digital audio streams,
- a GUI module interacts with the user via keyboard/mouse-input and monitor output,
- a MIDI module communicates with external MIDI-devices, such as a motorfader mixing desk or a MIDI-sequencer.

Other modules can include a disk-storage to save and load static presets or dynamic scenes, or composition specific algorithmic control units.

All modules should share the control data and be synchronised via a control bus, which internally can utilise the IP-protocol to distribute data between different threads and/or machines. (see Figure. 01)
In order to add as little overhead to an IP-based communication between modules, it is advisable to use a protocol that is based on UDP, a lightweight and fast, though stateless and connectionless internet protocol. A good candidate for this is OpenSoundControl (OSC), which is rapidly becoming the standard for network-based communication between audio software [03]. One advantage of such a network-based control bus is, that it allows the entire mixing environment to be remote controlled by the artist's favourite software for interactive/algorithmic structure generation, e.g. SuperCollider3 [04], Max/MSP or Pure data [05].

4. The Mixer (DSP module)

From a user's point of view, it is important to be able to intuitively handle such a system. Since the target users are already used to traditional mixer concepts, they should be able to also use this knowledge in a periphonic mixing environment.

The element of a spatial mixer that resembles a traditional multi-track mixer the most is the "source channel".

Since all routing is done in software, it is possible to assign an arbitrary source, like soundcard input, soundfile-players and generators as well as submixes of other channels, to each channel at no additional cost. Also, a single source can be used as input for several channels.

Effect-processing of the signal is very much dependent on the piece to be performed: a general "plugin" mechanism allows to adapt the signal processing of each channel. A small number of "standard" processing techniques (such as equalization, delay, inversion) might be implemented directly. While the channel signal is routed through a plugin, it does not necessarily have to change the signal. Instead this mechanism can also be used to modify control values for a single channel (or a number thereof).

To keep the channel setup as flexible as possible, each channel is considered to be mono. If stereo- or other grouping of channels is needed, this can be achieved via special plugins that duplicate the control values to other channels.

4.1. Spatialization

The final stage of each source channel is a spatialization block, which routes the signal according to the spatialization type of the channel to one of several modules.

For traditional mixing a "BUS" module mixes each channel to one or several busses. Each of these busses can be assigned to any group of output-channels via a matrix.

For abstract periphonic mixing, an "ambisonic encoder" module can encode the channel into an ambisonic soundfield, depending on the position of a virtual sound object. For concert performances, this soundfield would then be decoded to a periphonic loudspeaker array. Since such a loudspeaker array is usually not available in production environments (studios), the soundfield could also be decoded to binaural signals using the "virtual ambisonics" approach [06]. In this case, the ambisonic multichannel signal is first decoded to an idealized virtual loudspeaker setup. The virtual loudspeakers are then reproduced to headphone signals by filtering the feeds with head related transfer functions (HRTFs).

4.2. Extensions

To provide a possibility to add special processing units which are not bound to the concepts channels, a general "extensions" mechanism is suggested. Such extensions could implement alternative spatialization modules, reverberation algorithms or multitrack soundfile recorders/players to directly store or playback entire soundfields.
4.3 Virtual Acoustics

A straightforward way to add virtual acoustics is a digital reverberator using feedback delay networks. By applying such a reverberation to ambisonic encoded soundfields, it is possible to create periphonic acoustics.

The drawback of this method is that it is usually only applied to the music. This does not correspond to our experience in the real world, where also the audience is within the acoustic spaciousness. This shortcoming can be overcome with systems like the "ambisonic room in room reverberation (ARRR)".

The idea of this algorithm is to superimpose a given acoustic space (the concert hall) by another acoustic space (e.g. a cathedral). The room is recorded via a microphone array. Digital reverberation is added as needed. The reverberated soundfield is then fed back into the room via virtual reverberation sources. By smart placement of these reverberation sources within the reproduced soundfield, it is possible to give the reverberation a directional characteristic. To avoid unwanted feedbacks the microphones should be placed as far away from the corresponding reverberation sources as possible. Slightly wobbling the entire soundfield helps to suppress strong room-modes as well as selective filtering.

5. Implementation

An implementation of this environment has been done on a PC based system, using Miller Puckette's graphical realtime audio environment "Pure data". The mixer application runs on an AMD64 dual core processor. Audio interfacing is done via an RME Hammerfall DSP MADI card, providing 64 digital in-and output channels. A 16 channel motorfader console is used for haptic control of the mixer.

6. Conclusions

In this paper we presented a concept for a modular periphonic mixing environment, suitable for the reproduction of contemporary electro-acoustic music. The system has been successfully deployed during several concerts, both within specially equipped concert hall and outside, demonstrating that it can be used in a variety of locations. Music performed with this system includes classic multichannel tape-pieces as well as pieces using complex real-time algorithms, proving that it is modular enough to match the needs of a wide range of target applications.

7. References


