In the past, we have reported on efforts to create a comprehensive set of studio production tools for the NeXT computer and the Ariel/IPCAM ISPW board [Otto et al., 1992]. The recent decision by NeXT to discontinue hardware manufacture, and by extension, the definitional obsolescence of the ISPW board, has caused us to reconsider the path along which MixNet had been developing. These circumstances have given us the opportunity to formulate a long-range vision for audio production methods in the future, and to begin to lay a groundwork for the eventual integration of audio production software tools into a more comprehensive suite of media (video, CD-based photography, CDI, audio, telepresence, teleconferencing, virtual reality, etc.) production and manipulation applications.

One idea which seems to be gaining currency among developers and institutions concerned with the interoperability of diverse technologies is embodied in the emerging CORBA (Common Object Request Broker Architecture) standard [Soley, 1993]. Programs which are CORBA compliant would have the ability to query remote devices on the network as to the set of functions (services) provided, data types returned, and arguments expected. In addition, a CORBA compliant program would have the ability to advertise its own services to remote devices, as these functions came on line.

Under this scheme, MixNet itself evolves into a set of GUI tools and palettes for constructing custom processing algorithms, as well as for more or less well defined functions such as multi-track automated mixing, recording, soundfile editing, etc. MixNet, however, would be only a front end, and would communicate with a remote DSP server (the server could in fact be running in the same box, as a separate and concurrently running process) to order specific services, including analog and digital I/O, to be performed on specified data. The DSP server could be fTS (the server
underlying MAX on the NeXT) [Puckette, 1991], or it could be a different design altogether. We envision an intelligent server which is capable of performing automatic load balancing on multiple processors, pre-empt processes according to an established priority scheme, communicate process and request status to the user, and which has the ability to reconfigure signal paths on the fly without interrupting existing processes. In addition, this scheme would allow other parties to supply alternative front ends which access the same DSP server. One can thus imagine a system (such as might be located in a large institution) that contains several processing servers (for audio, video, and other media), various user stations on mixed platforms, all connected by a high bandwidth network, and able within reasonable limits to communicate both control data and signals amongst themselves.

Clearly for such a system to be truly effective, standard protocols for the transmission of data between devices, as well as the nature of the services provided by devices on the network, would have to be established. We especially welcome any suggestions and input that others may have on this matter.

On a related subject, we have also been experimenting with the notion of the reconfigurable console. The illustration on the next page is one manifestation of this idea, wherein components from a console strip (faders, EQ, panners, sends, effects, etc.) are copied from a palette window (not shown) and placed at will within a virtual console strip. This allows the creation of arbitrarily defined signal paths and the explicit allocation of available resources. This console is currently running as a client on top of FTS.

References


In the virtual console above, strips 1 and 2 contain the same components, but strip 2 has been scrolled down in order to expose the EQ components at the top. Auxiliary send components have been added to strips 3 and 4, and strip 5 contains three EQ components and a fader, but is missing panners, solo, mute, etc. Components may be placed in any desired vertical order within the strip. This order determines the signal path, making, for example, the difference between a pre and post fader EQ immediately apparent by position. Each component has a dimpled button at the top right corner which is used to "pop the hood," to determine details such as functionality, range of operation, control by external hardware devices, or control input for each component. Additional channels may be added simply by resizing the window — the addition of more surface area automatically results in the creation of new strips. The window may also be resized vertically to expose the entire strip, or at least as much as will fit on the screen.