UNISON - A REAL-TIME INTERACTIVE SYSTEM FOR DIGITAL SOUND SYNTHESIS

John A. Tate
Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
email: Tate@ccm.umanitoba.ca

ABSTRACT: This system allows sound synthesis algorithms to be designed and entered by connecting graphical representations of modules on a computer screen. The results are synthesized in real time using a digital signal processor connected to the host computer. The system is interactive, allowing the user to change parameters at any time and hear the results instantly. Graphical changes to the algorithm on the screen also affect the system with a minimum of delay, and without user intervention. New algorithms may be encapsulated to form additional modules which may then be used to create more complex algorithms in a hierarchical manner.

Background:
One of the major attractions in using a computer to produce sound is that there are essentially no limitations on the algorithms and techniques which can be used. Software synthesis programs such as Music V take advantage of this flexibility, but lack the immediate response of dedicated fixed-algorithm synthesizers. By using high-speed digital signal processors, it has become possible to perform a certain amount of sound generation and modification in real time, thus removing one of the major drawbacks of software synthesis. However, one must still program the signal processors. This can be time-consuming and difficult when it is done by traditional methods and again there is a lack of immediate feedback. By using the graphical user interface now common on most computers, it seems reasonable to allow the user to work at a higher level in which algorithms are specified simply by connecting graphical representations of the desired modules or operations. The resulting diagram may then be compiled into code for a signal processor, which will produce the final audible result. Several such systems have recently been developed [1,2,3].

UNISON is an integrated environment in which a graphical user interface, a compiler, and a digital signal processor are closely linked to form a sound synthesis system which is flexible and interactive, and which operates in real time. The user is given a set of devices which perform basic functions such as waveform and envelope generation, filtering, arithmetic operations, delays, etc. These devices may be interconnected graphically to form a circuit which is compiled into the corresponding code for a Motorola 68001 digital signal processor. Whenever significant changes are made to the circuit, the code is automatically recompiled and loaded so that the results may be heard very quickly. In addition, devices may contain controls which affect their operation. For example, an oscillator may have controls affecting frequency and amplitude. These controls may be manipulated on the screen by the user, and their effect is effectively instantaneous. The adjustment of synthesis parameters, mixing levels, and the like will therefore have immediate and audible effects.

Physical Description:
UNISON is primarily a software product which runs on most Apple Macintosh computers. In order to actually produce sound, a Digidesign Sound Accelerator card is required. This is a commercial product which is available in versions for the Macintosh II, SE, and SE/30. It provides a Motorola 68001 digital signal processor as well as a small amount of the high-speed memory necessary for its operation. The Macintosh is able to communicate with the Sound Accelerator fairly quickly using the NuBus, enabling high-speed transfer of signal processor code and data.

ICMC GLASGOW 1980 PROCEEDINGS
Devices:
A device is an object which performs some operation or set of operations related to sound production. Each device has an associated picture which determines its appearance on the screen, as well as one or more input or output pins which represent the digital signal streams entering and leaving the device. There may also be controls in the device which allow the user to affect its operation in certain ways.

A set of primitive pre-defined devices is supplied, a few of which are shown below.

Device (A) is a sine oscillator with inputs controlling frequency, amplitude, and phase. Device (B) is another sine oscillator containing two controls which allow the user to select a fixed frequency and amplitude. Device (C) is a MIDI input device with outputs for frequency and amplitude (note velocity) as well as a trigger (note on/off). It contains a control which allows the user to select the desired MIDI channel. Device D is an envelope generator. Clicking the mouse in the rectangle containing the envelope symbol will bring up a larger window containing controls which determine the envelope shape. Devices (E) and (F) perform simple arithmetic operations. Devices (B), (C), and (D) also contain name controls which allow the user to label them.

Circuits:
Devices are interconnected to form circuits. For example, in order to experiment with simple frequency modulation, the user might draw the circuit shown below.

Sound output will be heard within a second or two of drawing this circuit on the screen, and changes to the four controls will produce immediate audible results. Modifications are also easily performed. For example, the addition operation in the above example could be quickly replaced by a multiplication operation, yielding a completely different result.

Circuit Compilation and Operation:
Associated with each type of device is a block of 56001 machine language code, together with a set of required linkage information. There are also descriptions of internal data memory requirements, controls, and input and output pins. When a circuit is created, blocks of code corresponding to the devices used are concatenated to form a larger block of code which will be executed by the signal processor once for each output sample. Every set of one or more inter-connected pins in the circuit is assigned a memory location in the data space of the signal.
processor, and additional space is assigned to satisfy the internal requirements of each device. A simple linkage editor is then used to patch the code in order to resolve all data references.

Each control in the circuit will affect one or more of the internal memory locations or I/O pins of the device containing it. Each of these will have an associated address in the data memory space of the signal processor so the circuit has been compiled and linked. When the user manipulates a control, or when MIDI data is received by the host computer, the appropriate values are simply stored into the corresponding addresses in the signal processor’s memory.

Device Creation:

New devices may be created in two different ways. The user can create a new primitive device by directly supplying the signal processor code, the linkage information, and description of the picture, pins, and controls. A simple symbolic format is provided for entering this data. However, the user must be capable of writing, assembling, and debugging 56001 code in some other environment before presenting the results to UniSon. This is not a practical choice for most users.

Any circuit may also be encapsulated to form a user-defined device. The compiled and linked code for the circuit will become the code for the new device. Special devices are provided which enable input and output connections to be specified and named. Any controls that are present in the circuit may be inherited by the new device formed from that circuit. The user may create an appropriate device picture using any of the standard drawing programs. For example, by adding two envelope generators, a MIDI input, and a few more arithmetic operations to the simple FM circuit given earlier, a more useful FM “voice” could be created and made into the device below.

![Device Diagram]

This device contains seven controls: three sliders controlling carrier and modulator frequencies (relative to the frequency of the incoming MIDI note) and modulator amplitude, two envelope controls, a MIDI channel selector, and i name. This device will respond to the specified MIDI channel and produce a simple FM voice with the specified parameters. It may be used to create circuits which in turn may define more complex devices.

Using this capability, one can imagine building up an entire studio environment completely with synthesis modules, mixers, sound processors, and so on. Unfortunately, complex devices such as these will very quickly exhaust the capabilities of any single signal processor. In order to create a complete and fully equipped “studio is a box” using this software, rough calculations indicate that a processor would be required with a speed in the range of a billion operations per second. Actually, such hardware will likely become affordable in the not too distant future. In the meantime, UniSon may be used to investigate new and flexible synthesis algorithms in a responsive interactive environment. It is an ideal educational tool, allowing synthesizers, filters, and digital effects to be created, modified, and studied in a simple and intuitive way.

References:


ICMC GLASGOW 1990 PROCEEDINGS