MAX+UniSon – Interactive control of a digital signal multiprocessor

John A. Rate

Department of Computer Science, University of Manitoba
Winnipeg, Manitoba, Canada R3T 2N2
e-mail: rate@cs.umanitoba.ca

ABSTRACT

UniSon is an interactive real-time system for creating and controlling digital synthesis and signal processing algorithms using a graphical user interface. It runs on a Macintosh computer using a Sound Accelerator, Audiomedia, or similar 56001-based NuBus card. A custom-built digital signal multiprocessor has been used to expand the amount of processing that can be done in real time in a manner that is transparent to the user. In this paper, I describe a version of UniSon which uses inter-program communication to allow higher-level control functions to be provided directly by Opcode's MAX program. This provides functionality similar to that of the "slide objects" (DSP objects) in the version of MAX running on the IRCAM IMW machine.

1. INTRODUCTION

UniSon combines a graphical user interface, a compiler, a linkage editor, and a DSP interface to create an environment in which real-time sound synthesis and digital signal processing algorithms may be created, used, and modified interactively. One of the principal UniSon windows is shown in Figure 1. Each of the interconnected blocks in this window is a device which performs a particular function. Each device has a number of I/O pins and may also contain controls which allow the user to affect its operation. The devices in Figure 1 contain scroll bars, envelopes, and an on/off switch. Circuits are formed by connecting a set of devices using lines each of which represents a 44.1 kHz digital audio signal. New devices may be created by encapsulating circuits in which case the device may inherit any of the controls present in the circuit, forming a type of control panel. Devices may also be created by directly supplying 56001 DSP code together with the required linkage information. The synthesis is done in real time, and interactive control of all parameters is possible. Whenever a significant change is made to a circuit, it is recompiled into a block of DSP code, re-linked, and re-loaded into the Sound Accelerator. For practical circuit sizes, this happens in a fraction of a second. Whenever the user manipulates one of the controls, updated parameter information is sent to the DSP using a fast interrupt mechanism which makes its effect almost instantaneous. By using a multiprocessor card with a specific interconnection architecture (Rate 1991), fairly large circuits can be handled with no additional effort on the part of the user. For more details, please contact the author directly at the address above.

![Figure 1. A circuit window from UniSon](image-url)
2. MAX-to-UniSon COMMUNICATION

UniSon has relied on MIDI for overall control. However, since MAX is a somewhat similar program which operates on higher-level event-based information, it seems natural to combine the two programs in order to provide similar flexibility at all levels of operation. Such functionality is provided in the MAX implementation on the IRCAM IMW machine described by Puckett (1991). In that system, the signal processing network is constructed directly in the MAX environment using special “wire objects” which perform DSP-based functions. However, it seems reasonable to have two separate programs to handle the two very different types of processing. The connecting lines in MAX allow discrete high-level messages to be interchanged in flexible patterns, whereas the connecting lines in UniSon represent continuous 44.1 kHz digital signal streams.

The communication between MAX and UniSon is designed to be as simple as possible from the user’s point of view. A sample MAX window is shown on the left in Figure 2. A new external object with the name “UniSon” is used to handle all of the necessary tasks. This object accepts a single symbolic name as its parameter, and has a single inlet. When a “connect” message is received at the inlet of any such device, it causes a Program-to-Program Communications Toolbox (PCT) port to be created and also opens a session between MAX and UniSon. The two programs may be running on the same Macintosh, or on separate systems connected by a network. The “disconnect” message terminates the session and closes the port. While a session is in progress, any integer message received at the inlet of a “UniSon” object will immediately be sent to the device with the corresponding name in the currently active UniSon circuit. On the right in Figure 2 is a sample circuit window from UniSon containing two “MAX” devices which set up two communication paths using the names “frequency” and “gate”. The MAX patch accepts MIDI note messages and translates them into a frequency value and a gate signal which UniSon then uses to synthesize a simple sine tone with an amplitude envelope. A small amount of translation is needed to provide the proper data format for the 56001 signal processor (which uses 24-bit fixed point reals). The range 0-8388608 in MAX corresponds to the range 0.0-1.0 in UniSon. Values greater than 0.5 will trigger the envelope in UniSon, which accounts for the unusual-looking value 4164304 (0.5) in the multiplication object in the MAX window. A table is also required to translate the MIDI note numbers preferred by MAX into the corresponding frequency in the range required by oscillators in UniSon (0.0-1.0 or 0.0-22050 Hz). Whenever a change is made to the UniSon circuit, a new table of names is sent to MAX, together with a version number. MAX must then search the table for the names that appear in the “UniSon objects”. Thereafter, integer messages in MAX are passed to UniSon very quickly, at the interrupt level, using routines that are often a few statements long in each program.

Figure 2: Sample windows from MAX (left) and UniSon (right)

REFERENCES

357