NEW LANGUAGES FOR PHYSICAL MODELING SYNTHESIS

Benjamin Schroeder, Marc Ainger, Richard Parent

The Ohio State University

1. INTRODUCTION

The first great computer music language was Music V [2]. It set the stage for the next forty years: nearly all of today's computer music languages are philosophical descendants of the Music N family.

This family of languages is organized around the paradigm of a unit generator network. A unit generator provides a stream of audio samples; simple examples are a sinusoid generator and a generator which adds two signals together. Individual generators are kept simple and fast; waveform generators were originally implemented as flexible table lookups. This supports a method of producing sound by means of operations on other sounds: filtering, splitting, recombining.

The unit generator paradigm has been very successful, partly because it is flexible and partly because synthesis using unit generators can be (relatively) computationally inexpensive. However, it can be difficult to reproduce a wide variety of realistic playing techniques and instrument responses using this kind of synthesis.

2. PHYSICAL MODELING

The time is thus ripe to ask: do we need a new approach to the computer music language? Physical modeling presents one possible alternative. Although physical models can be represented to a certain degree in the Music N paradigm, that paradigm is not optimal for the expression of the full range of physical modeling techniques.

Physical modeling aims to recreate the way a sound is produced, with the sound obtained as the result of simulating a physical process. This is in a sense more complex than other kinds of synthesis but also much more expressive. Different styles of playing, for example, may be described directly, in terms of forces and players' actions, rather than in terms of the resulting sound. New instruments may be created which retain realistic qualities despite their novelty. Models may be related to one another in 3D space. Physical models have the potential to be as expressive as real instruments and yet as flexible as the computer. [1]

All of this is not without cost. The simulation of physical models is in general computationally expensive. Physical models have been studied for many years, but have often been thought too expensive to use, especially for live performance. Recent advances in computer speed, though, have made it possible to use physical modeling in many real time situations.

Physical model simulation is different than synthesis based on signals, in both description and implementation. How should we describe physically modeled instruments, and how should we play them? What language structures can optimally express this approach?

3. DISCUSSION

Consider as a first example a vibrating guitar string. To pluck such a string one might want to consider the location of the pluck, the strength of the pluck, whether the guitarist uses a pick or the nail (and how much nail); all of these have significant effect on the resulting sound. These differences could be represented as parameters to the string model.

Now imagine that the string is not only plucked, but is also fretted in the course of playing several notes in a tune. The string model remains the same, but needs additional parameters to represent a second interacting object (the left hand). To accurately model note-to-note transitions, we might want several sets of parameters representing several left-hand fingers moving across multiple strings.

With this in mind, we might go beyond simple parameterization to think of a language in which different objects could interact through forces which vary arbitrarily in time and space. This kind of abstraction would also be useful for concerns such as coupling a string to a resonating body.

Output is another area where new language constructs may be of use. Many physical models can provide output from any continuous location or even from an integration over the entire body. This suggests new structures for describing output, whether via virtual pickups and microphones or through situating the model in space.

Many other questions arise, about topics ranging from modeling to timing to notation and graphical output. We are asking these questions and invite you to join us.

4. REFERENCES

[1] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. Natick, MA, USA: A. K. Peters, Ltd., 2002.

[2] M. V. Mathews *et al.*, *The Technology of Computer Music*. The MIT Press, 1969.