Chapter IX
Driving Sound Synthesis with a Live Audio Signal
Cornelius Poepel
Academy of Media Arts, Cologne, Germany
& University of Birmingham, UK

ABSTRACT
An overview on problems and methods to map performers’ actions to a synthesized sound is presented. Approaches incorporating the audio signal are described, and a synthesis method called “audio signal driven sound synthesis” is introduced. It uses the raw audio signal of a traditional instrument to drive a synthesis algorithm. The system tries to support musicians with satisfying instrument-specific playability. In contrast to common methods that try to increase openness for the player’s input, openness of the system is achieved here by leaving essential playing parameters nonformalized as far as possible. Three implementations of the method and one application are described. An empirical study and experiences with users testing the system implemented for a bowed string instrument are presented. This implementation represents a specific case of a broader range of approaches to the treatment of user input, which have applications in a wide variety of contexts involving human-computer interaction.

INTRODUCTION
Today’s methods of sound synthesis, developed by technical progress over the last decades, arguably provide an almost inexhaustible potential for the performance of music. Thus, it would seem that what has traditionally been a task of finding acceptable solutions amongst limited possibilities has become the opposite. Today, the producer is faced with the challenge of finding the exceptional amongst a supposedly unlimited reservoir. The limitations have thus altered. For the performer they exist less in the potential of the material rather than in its availability. Regarding quantity of sonic possibilities and spectral variety, one can say that computer-based instruments outdo acoustic instruments by far. Regarding their usability for musical expression, however, due to
limits in the interfaces, they are still lacking in some points (Fels, 2004).

Research on musical instruments and interfaces falls into the field of human-computer interaction, which is represented by conferences like the Conference on Human Factors in Computing Systems (CHI, 2007). A growing community working in the field of new interfaces for computer-based instruments has evolved in recent years. One important place to present outcome of research and development can be found in the international conference on New Interfaces for Musical Expression (NIME, 2001). The NIME conference evolved in 2001 out of the CHI conference. Many new interfaces have been presented there.

In contrast to the widely used MIDI keyboard, the majority of these new developments have not been commercially successful. Concerning the keyboard, one might argue that two factors were of importance for its success. Since its interface is based on the piano, there were always performers who could use existing skills to play the instrument. In addition, there was a huge body of repertoire that could be used.

Trying to replicate the success of the keyboard with other instruments like woodwinds or strings was not that effective due to the specific needs a musical interface has to meet in order to satisfy a musician. The keyboard emphasis of the MIDI protocol made it particularly difficult to adapt non-keyboard instruments as controllers.

An increasing number of researchers are addressing the questions around musical interfaces, and how the needs of musicians might be successfully met in their creation (Wanderley & Battier, 2000). Desired qualities of musical interfaces are described with terms like the “playability” of an instrument (Young & Serafin, 2003), its “feel” (O’Modhrain, 2000), its “intimacy” (Fels, 2004), or its “transparency” (Fels, Gadd, & Mulder, 2002).

Common to attempts to increase the quality of computer-based instruments is the assumption that such qualities can be realized through work on either the interface, or on the mapping method between interface and the synthesis engine. It is also common to use synthesis engines that are driven by explicit input parameters, and to control the engine with measured input that attempts to describe the actions of the performer.

According to statements of Buxton found in Buchla et al. (Buchla, Buxton, Chafe, Machover, Moog, Mathews, Risset, Sonami, & Waiswiz, 2000), a huge body of literature concerning musical interfaces has been provided in the last years without a concomitant effect on the interfaces the majority of musicians are using. This raises the question of whether it might not be helpful to study and to modify the general architectural principles of the interplay between interface and synthesis.

A main principle in building computer-based instruments has been the definition of a fixed set of formalized and measurable playing parameters derived from the actions of the performer. Therefore, studying performers’ actions is crucial (Wanderley & Depalle, 2004). However, in comparison to this, an acoustic instrument, like a traditional violin, has no explicit formal inputs. While it is possible to define and measure those in the performer-instrument interaction from a physical point of view, it remains unclear whether this may or may not represent completely what the player does.

Taking any given formalisation of an interface based on a physical representation of the performer-instrument interaction, it arguably remains unclear whether a performer is able to make use of the full range of techniques for the creation of musical expression that she or he is used to having at her or his disposal. In order to solve the problem of such potential limitations, it is worth considering the possibility of design principles that can deal implicitly with playing techniques in a manner analogous to the way acoustic instruments do.