Editor’s Notes

Computer music has arrived. From the field’s inception, we had to struggle with technology just to make our music happen. Out of necessity, our pioneers developed some of the earliest mixing consoles to diffuse sound in space, devised vacuum-tube digital-to-analog converters to hear the very first digital sound samples, created efficient synthesis and digital signal processing algorithms to manipulate sound in real time, designed low-latency hardware and software in order to coordinate multiple sound-making devices—the list goes on. But we have now won most of these battles, and have entered an age in which many of our fundamental challenges are no longer necessarily technological ones; rather, we are faced with the questions of how to effectively and creatively organize, assemble, and employ the dizzying array of technologies now at our disposal.

We can draw an interesting parallel to the domain of mainstream office computing. By the mid 1980s, the technologies upon which the desktop computing paradigm is based had largely codified; indeed, the desktop office computer of today, with all of its peripherals, does not look much different than it did then. And it was around that time that the field of human–computer interaction [HCI] burgeoned in earnest. Questions of how best to wrap these technologies into novel products to yield the most efficient, reliable, and satisfying results came to the forefront. HCI gave us this new perspective, as well as sets of methodologies, frameworks, and practices with which to go about addressing these questions.

Being more technologically demanding, the field of computer music has taken a bit longer to reach a similar point in our history, but we have arrived. Of course, plenty of technological work remains to be done; however, many of the platforms we use have largely stabilized, and inexpensive, readily available software and hardware components are sufficiently fast and reliable to accomplish much of what we want to do. But unlike mainstream office computing, it is only in the past five to ten years that we have been able to say with some confidence that we largely have all the widgets and doodads we need. And it is precisely in that time that researchers and practitioners have begun to be concerned, in significant numbers, with issues of HCI that are specific to music creation. This new focus is evident in the dramatic rise of the Conference on New Interfaces for Musical Expression [NIME], which has just marked its tenth anniversary, and indeed in the diversity and quality of the articles in this special issue of Computer Music Journal, not to mention an overwhelming number of strong submissions that could not be included.

This issue begins with an interview conducted by Gascia Ouzounian with Paul DeMarinis, the keynote speaker at NIME 2009 at Carnegie Mellon University. Paul DeMarinis was himself an innovator in the technologies and techniques of interactive music and sound, but his work also often celebrates the early pioneers of science and technology—known, obscure, famous, and forgotten—situating their visions in fanciful and poignant contexts. His art twists technology as subject and object, it is as much about technology as it is of technology, and in the spirit of this issue it prompts us to reflect on our relationships with the fundamental forces and media on which much of our work relies.

NIME has been largely concerned with hardware interfaces and physical interaction, to the extent that it seems software is sometimes neglected. Two articles in this issue therefore came as pleasant surprises, as they investigate interactions with music software in two different contexts, and with two contrasting approaches drawn from HCI. The first, “Abstraction and Activity in Computer-Mediated Music Production” by Matthew Duignan, James Noble, and Robert Biddle, draws on activity theory and Blackwell and Green’s cognitive dimensions framework to describe the actual activities of composers and producers using music production software, to identify the extent to which these tools facilitate the outcomes the musicians are trying to achieve, and to suggest ways in which the software could better support these activities.

Also applying established theories from HCI to interactions with music software in “What Can the Language of Musicians Tell Us about Music Interaction Design?” Katie Wilkie, Simon Holland, and Paul Mulholland examine the ways in which people map generic image schemas, such as containment, spatial orientation, and balance, onto musical concepts to create the conceptual metaphors we use to understand music. The authors analyze two software programs in terms of the image schemas they suggest or support, and discuss the relationships between diverse image schemas or conceptual metaphors and the specific purposes or goals of the software. This article primarily focuses on how and what we know when experiencing music, forming a nice contrast to that of Duignan, Noble, and Biddle, which looks at how and what we do when creating music.

In “Performing Musical Interaction: Lessons from Interactive Theatrical Experiences,” Steve Benford details some of the frameworks and theories that emerged from his groundbreaking work studying interactive theatrical performances, reexamining them through a musical lens. This article shifts the issue’s focus onto interactivity in a performance context, and also exemplifies another
aspect of contemporary HCI—the use of ethnographic methods and field work with roots in sociology—demonstrating how such approaches might play a greater role in the study of musical interaction.

Where the previous three articles show how established methods, frameworks, theories, and techniques of HCI can be successfully leveraged and applied in investigating musical interactions with computers, in some instances the unique nature of our field requires a divergent perspective. Thor Magnusson makes this case in his article, “Designing Constraints: Composing and Performing with Digital Musical Systems,” arguing that where HCI would typically be concerned with affordances, the notion of constraint is often a more useful design principle in interactive music systems.


It was a joy and a privilege to put this issue together, and heartening to see so much diverse activity in this area all around the world. I hope readers enjoy these articles as much as I did.

Michael Gurevich, Guest Editor