Published as: Roger B. Dannenberg, “Some Histories and Futures of Making Music with Computers,” in *Korean Elecro-Acoustic Music Society’s 2020 Annual Conference Proceedings*, 2020, pp. 7-14.

Some Histories and Futures of Making Music with Computers

Roger B. Dannenberg

Carnegie Mellon University
rbd [at] cs.cmu.edu
http://www.cs.cmu.edu/~rbd/

Proceedings of Korean Electro-Acoustic Music Society's Annual Conference (KEAMSAC)2020
Seoul, Korea, 30-31 October 2020

Having spent decades working in the field of Computer Music, I review some major trends of artistic and scientific development in the field with an eye to the future. While the implications of exponential computational growth are hard to predict, it seems that musical imperatives remain stable; thus, they will continue to guide us in the future. I predict a number of “futures” for Computer Music based on the persistent themes of sound creation, music representation, and music performance.

Keywords: History of computer music, Futures of making music with computers, Moonshot project.

Like computing itself, Computer Music has experienced rapid growth over sixty years or so. We have seen an evolution starting from primitive but pioneering attempts to create the first digital musical sounds and to create and control music algorithmically. Our current state-of-the-art now includes very sophisticated real-time signal processing, flexible software languages and architectures, and commercialization that reaches billions of creators and consumers. I am honored to address the KEAMS Annual Conference 2020, and I would like to take this opportunity to look both backward and forward with an aim to better understand the field and perhaps to gain some insights into future artistic opportunities and scientific directions.

Most of my work in the field has been scientific, but I feel that my work has always been guided by my experience as a performing musician and composer. My early interests in math, music and engineering led me to analog music synthesizers as well as computers in my teens. (I should add that computers around 1970 were rarely encountered outside of businesses and universities.) Through college, I learned enough electrical engineering to design and build a hybrid digital and analog synthesizer as well as a microcomputer of my own design and wired by hand, but I was pretty ignorant of emerging research. At least I was well prepared to suddenly discover a small but growing literature from authors and editors such as Max Mathews, Jim Beauchamp, John Chowning and John Strawn. I spent my years in graduate schools in more mainstream Computer Science, but on the side, I devoured everything I could find to read on Computer Music. I emerged from graduate school with a junior faculty position and a very supportive, open-minded senior faculty including Nico Habermann, Raj Reddy, Alan Newell, Herb Simon, and Dana Scott. Ever since then, I have been very fortunate to follow my passion for Computer Music making and research. I have closely followed and participated in over four decades of Computer Music development.

In this presentation, I wish to review some of my own work, which like all research is tangled in a network of other ideas and influences. From this, I hope to draw some understanding of the big ideas that drive the field forward. The occasion of a keynote address is one of those rare opportunities where one can be controversial and speculative. I will take this chance to make some predictions of where we might be going in the future. I have titled this talk in the plural: both “histories” and “futures” to hedge my bets. There are multiple ways to organize the past and multiple possibilities for the future. And speaking of the title, the phrase “with Computers” is purposefully ambiguous, regarding computers as both tools and collaborators. I will surely omit some important history and fail to anticipate much of what is yet to come, but I hope these ideas might inspire some or at least offer interesting insights.

Why Computer Music?

Anything new, any break with tradition, is going to raise questions. For some, computers and music seem a natural combination – why not? For others, as if the pursuit of Computer Music detracts from something else, what is the point? I have been collecting answers for many years, although I think there are really just a few. One idea that I was introduced to by F. Richard Moore is the *precision* that digital computation brings to music. Instead of music where every performance is unique, computers give us the possibility of precise reproduction, and thus incremental refinement of sounds with unprecedented levels of control.

Another important idea is that composers, rather than create directly, can create through computational models of composition. This has two implications. The first is that computational processes can be free of bias, so just as a tone row might help to liberate a composer from tonal habits, a computer model might create new musical structures and logic that the composer could not create directly. The second implication is that composers can inject new musical logics or languages into real-time interactive performances. This enables a new kind of improvised music where performers are empowered to bring their expressive ideas to the performance, but computers can enforce the compositional plans and intentions of the composer. It is as if the computer program becomes a *new kind of music notation*, constraining the performer in some respects, but leaving expressive opportunity in others. In my view, this is a powerful extension of aleatoric writing, which prior to computing found only limited ways to split musical decisions between the performer and composer.

These rather technical rationale for Computer Music, important as they may be to justify our work, are really just excuses for us to do what we love to do. Humans have an innate fascination with technology and automation. As soon as you tell someone that a robot is involved, the story is immediately interesting. Experiments by my advisee Gus Xia, *et al*. (2016) give evidence to what I call the "robot effect:" Suppose a human performs along with an audio recording, as in mixed music performances. How can we make the performance more engaging for the audience? One approach is using interactive, responsive, automated computer accompaniment. This in fact does not help much. Another approach is humanoid robot performers playing a fixed score, as in animatronics, but this does not help much either. However, if we *combine* computer accompaniment with humanoid robots to create *interactive robot performers*, then the audience finds the performance more engaging and more musical! This is evidence that we *are* innately attracted to the automation of human tasks, and what could be more human than making music?

All of the ideas above combine with a basic urge to explore and learn. Do we really need an excuse or rationale? Let us pursue our passion and see where that leads. After so many contributions to the arts, science, and culture, we no longer have to worry whether we are on a good path. Let us now try to characterize the path we are on and where it might lead.

The Computer Music Dream

Taken as a field, Computer Music is following a path that reflects our general understanding of music. First, *sound* is a critical attribute of music. Thus, from the very beginning, Computer Music was about making sound, combining digital signal processing with digital computation to create musical tones. One could argue the first tones were hardly musical, but through many years of research, our capacity to create musical sounds surely surpassed even the wildest dreams of early researchers.

The second critical attribute of music is organization in time, exemplified by music notation. A great deal of early research concerned musical scores, note lists, music representation and music control. Just as sound synthesis has imitated the centuries-long development of acoustic instruments, music representation and control research has imitated centuries of development of music notation, from the development of neumes in the 9^{th} century and common practice notation, to graphic notations developed in the last 70 years or so.

Western music is assigns importance to both planning by composers and execution by performers, and thus music often has *two* characteristic representations: the score that represents instructions to performers, and live sound or recordings which convey the performance "product" to listener/consumers. (The same property holds for plays, film, and to some extent architecture and dance). Thus, a third thread of Computer Music research is an exploration and automation of performance, including interaction, expressive interpretation of scores, jazz improvisation, and performance style.

Although highly reductionist, I believe these three threads: *sound generation*, *music representation*, and *performance* serve to summarize our musical knowledge in general and also to describe the development of Computer Music.

The Impact of Technology

Throughout the history of Computer Music, the power of computers has grown at an exponential rate. It has been said that an order of magnitude difference is perceived as a *qualitative difference*, not just a numerical one, so we see a *qualitative difference* in computing every five to ten years. Each step through punched cards, time-sharing, personal computers, powerful laptops, cloud computing and mobile devices represents not just an increase in computing power but a new vista of opportunities for artists and researchers as well as a new framework within which we see problems and solutions.

Figure 1 illustrates growth in computing power over the history of Computer Music. The vertical axis is relative power, with a value of 1 assigned to the left-most year. The best measure of "computing power" is debatable, but all reasonable measures lead to the same conclusions. These graphs are purposefully plotted on a linear scale to show that, compared to today's computers in 2020, even computers from 2000 seem to have no ability whatsoever. Many believe the growth rate is slowing, so I have plotted the next 30 years with a doubling time of 3 years rather than 2, which is roughly the doubling time since 1960. The horizontal axis on the right is the same, but the vertical axis is reset to so that today's 1960's-relative computing power (2.5E+09) in the left graph appears as 1.0 in the right graph. As the graph shows, todays computers, which power Internet search, face recognition, life-like computer graphics and of course digital music processing, will seem completely insignificant by 2050. To get even a glimpse of what is in store for the next 30 years, consider that 30 years ago, software sound synthesis was barely possible. (Dannenberg & Mercer, 1992) Or consider that the release of our personal computer audio editor Audacity in 2020 was still a decade away. (Mazzoni & Dannenberg, 2002)

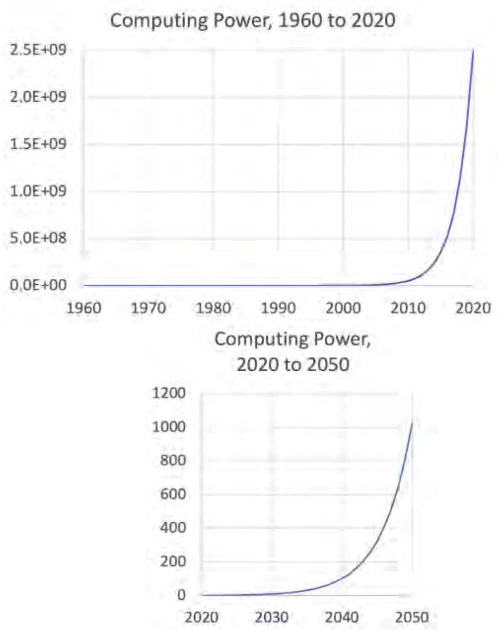

Figure 1. The growth of computing power has followed an exponential curve, doubling roughly every 2 years. Even if the doubling time slows to three years, today's computers will seem primitive within 20 or 30 years. The vertical axes represent relative power, with a value of 1 in 1960 (left) and 2020 (right).

One thing seems certain: We can imagine many developments in terms of today's technologies and devices, but the technologies of the future will be *qualitatively* different from what we have now. We will not continue to view problems in the same way. We can think about what we can do with faster computers, but it is much harder to imagine what new forms computing will take when computational power increases by orders of magnitude. We are probably better off to think in terms of musical imperatives.

A Brief History of Computer Music

To further explore these threads of sound generation, music representation, and performance, I would like to consider them in the context of some historical Computer Music developments. This is not meant to be a complete history by any means, but it will help set the context for thinking about possible futures.

Early Computer Music

In the earliest years of Computer Music, essentially all computers were mainframe computers that were programmed by submitting a stack of instructions on punched cards and receiving results in print or on magnetic tape. The first music sound generation software is exemplified by Max Mathew's Music *N* programs (Mathews M. , 1969), which already neatly capture the notions of sound and score (representation) in the "orchestra language" and the "score language." The former was designed to express digital signal processing needed to create *sound*, and the latter was a separate *music representation* language designed to express sequencing and control parameters for those signal processing operations.

Real-Time Digital Instruments

As soon as integrated circuits achieved enough power to perform basic audio signal-processing tasks in real time, digital instruments began to appear. Research systems such as the Dartmouth Digital Synthesizer (1973) and the Bell Labs Hal Alles Synthesizer (1976) led to commercial systems such as the CMI Fairlight and New England Digital Corporation Synclavier, which were soon followed by mass-produced instruments such as the Yamaha DX7 (1983). Viewed from the perspective of *performance* and the understanding of exponential growth in computer power, these developments were inevitable, even though keyboard instruments were *qualitatively* nothing like the programmed mainframe and minicomputers in use up to that time.

Interactive Systems

The combination of affordable real-time digital synthesis, the interface possibilities of MIDI, and the introduction of personal computers, all coalescing more-or-less in the 1980's, enabled a new direction in computer music: real-time musical interaction with computers. (Rowe, 1992) (Winkler, 1998) Many musicians developed interactive systems: Composed Improvisation (Eigenfeldt, 2007) by Joel Chadabe, The Sequential Drum by Max Mathews and Curtis Abbot (Mathews M. V., 1980), *Voyager* (Lewis, 2000) by George Lewis, Ron Kuivila's compositions with Formula (Anderson & Kuivila, 1990), and David Wessel's compositions with MIDI-Lisp (Wessel, Lavoie, P., Boynton, L., & Orlarey, Y., 1987) are just a few of many experimental works. In that time period, I designed the CMU MIDI Toolkit in 1984 (Dannenberg, The CMU MIDI Toolkit, 1986), inspired by Doug Collinge's *Moxie* (Collinge, 1985) language, and created *Jimmy Durante Boulevard* in a collaboration with Georges Bloch and Xavier Chabot (1989).

Interactive Systems brought compositional algorithms, previously only used for non-real-time composition, into the world of performance. Just as real-time synthesizers can be seen as joining digital sound and performance, interactive systems represent the union of music representation and composition with performance. As mentioned earlier, this created a new mode of composition. The composer specifies a piece not so much by writing notes as by writing interactions. These interactions continuously constrain and guide the sensitive musician to carry out the composer's plans. At the same time, the improviser is free to inject spontaneous and virtuosic elements that the composer might not have imagined. In the most successful work, a previously unknown and exciting synergy is achieved.

Computer Accompaniment

Another approach to interaction is based on the traditional model of chamber music where notes are determined in a score by the composer, but musicians perform the score with expressive timing. In the Computer Music world, composers were drawn to the possibilities of computation, which fixed music precisely in time, but the only way to combine that approach with live performance was to play along with a fixed recording. There was an obvious disconnect between using fixed media and the well-developed ideas of expressive performance in chamber music. In 1983, I began to experiment with algorithms, and I built a complete working accompaniment system in 1984 that could listen to my live trumpet performance, follow along in a score, and synthesize another part in real-time, synchronizing with the soloist. (Dannenberg, 1985) Similar work was introduced around the same time by Barry Vercoe. (1985) Later, my computer accompaniment work was used to create the Piano Tutor, an intelligent tutor for teaching beginning piano students (Dannenberg, et al., 1990), and computer accompaniment was commercialized in what is now SmartMusic and used by hundreds of thousands of students. Work on score following and collaborative performance is still an active topic today.

Human Computer Music Performance

Computer Accompaniment distilled the basic idea of following and score and synchronizing performance, but in music, there are many more problems related to collaboration. This came to my attention around 2005 when I was playing in a rock band's horn section. As the only trumpet, and not a strong lead player, I began to think how much better it would be if I were the *second* trumpet alongside a great high-note player. It did not take long to imagine I could use my computer accompaniment techniques to create a virtual musician for the band. However, I soon realized that the band did not always follow a score strictly from beginning to end. Also, horns do not play all the time, so how would the virtual player enter precisely in time and in tempo without following a leader? A virtual player might "listen" to the keyboard player, but the keyboardist improvises chord voicings and rhythms, so there is no detailed score to follow there.

These and other problems led me to think about musical collaboration much more broadly than before. Synchronization is achieved not only by following scores, but by following the beat, following chord progressions, visual cues, following conductors, becoming the leader, and combinations of these things. Parts are specified by traditional scores, lead sheets, drumming or percussion styles, and analogy ("I want you to play this part the way Bill Evans might do it.") In other words, the broader goal is not simply an "adaptive sequencer" that synchronizes to a pre-determined stream of notes, but an *artificially intelligent musical partner*.

We can see related work in laptop orchestras, networked music performance, and artificial intelligence for composition. These are all approaches that use technology for human-human and human-computer *music collaboration*.

Interlude

Let us try to sum up some ideas of this brief discussion. Computer Music has ridden a wave of exponential growth in computing power to get us where we are today. Much of our progress could never happen without integrated circuits, powerful computers and the whole information age (for example, only the pervasive adoption of computing in daily life could drive down price of billion-transistor processors to affordable levels.) However, the main directions of Computer Music can be seen as an attempt to reproduce and then extend traditional music concerns in three areas: sound, music representation, and performance.

We have discussed an historical progression in which researchers explored the production of sound, music representation and control, real-time interaction, computer accompaniment, and collaboration in general. The future will bring unimaginable computing technologies and with it multiple qualitative changes in the way we think about or experience computing. However, our principle musical concerns are likely to be the same ones we have pursued for centuries if not millennia, so with that assumption let us consider some implications for the future.

The Future of (Computer) Music

One way to conceptualize the whole of musical concerns is illustrated in Figure 2. Here we see "Instruments" as the world of sound generation and processing. While instruments produce sounds, musicians organize sounds into phrases, and there is much work to be done to understand phrases (more on this below). Phrases (or, in some terminologies, "musical gestures") are assembled to form compositions. Compositions are performed, giving rise to many concerns of collaboration and coordination. Let us consider each of these realms separately.

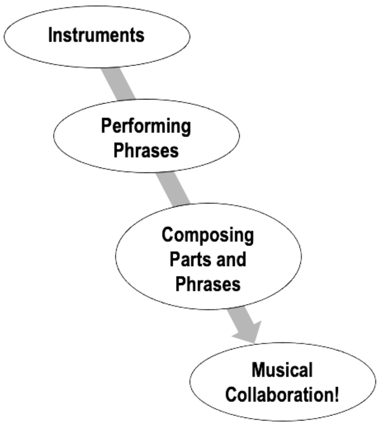

Figure 2. Schematic of Computer Music areas of concern.

Instruments

Even after decades of research, instrument modeling remains elusive. The non-linear, 3-dimensional physics of acoustic instruments are complex (Bilbao, 2009), and our perceptual abilities are exceptionally refined, making even slight imperfections quite apparent. Musicians take many years to learn to control acoustic instruments, and without control, even real acoustic instruments do not make interesting musical tones. It seems that in the future, orders of magnitude more computation will be applied to acoustic instrument simulation as well as to machine learning to discover how to control them to produce musical results. From there, new possibilities will emerge to artistically manipulate "physics" in our simulations to design new instruments and new sounds, informed but not limited by real acoustics. Spectral synthesis models based on computational models of perception are also a promising direction for new sound creation.

Another interesting direction is physical robotic instruments such as those explored by Trimpin, Eric Singer, Ajay Kapur and others. I helped Ben Brown and Garth Zeglin construct a high-performance robot bagpipe player, McBlare, at Carnegie Mellon University. (See Figure 3.) The "robot effect" described earlier suggests that we should pay attention to robots, and just as musicians have been able to use computers and sensors developed for other applications, I expect humanoid robots created with other purposes in mind will offer very engaging modes of musical performance.

Figure 3. McBlare, Carnegie Mellon's robotic bagpipe player.

Phrases

Many years ago, the mantra “sampling is dead” was frequently heard among computer music researchers. The basic idea of samples is to record “notes” of instruments and play them back on demand. If a violin plays a range of 4 octaves at 10 different dynamic levels, that is about 500 sounds, assuming we can find reasonable ways to control duration and simulate vibrato. In the early days of limited memory, even 50 very short samples that required “looping” to extend them was already expensive, so it seemed hopeless to achieve high quality through sampling. Over time, however, memory prices came down, so sample libraries could add longer samples and many variations of articulation, bow position, and even extended techniques. It seems that our predictions were premature.

However, expressive continuous control is still a problem for samples, and here is where *phrases* enter the picture. My work in the 90’s showed that the details of individual notes are highly dependent upon context. (A Study of Trumpet Envelopes, 1998) For example, a slurred transition between two trumpet notes is entirely different from an articulation where the air is briefly stopped by the tongue, and details of the transitions are also affected by the pitches of the notes. Thus, *phrases* are critical units for musical expression and even timbre, yet they have been largely ignored.

In the future, either sampling will have to “die” or expressive phrases available to string and wind players will disappear from electronic music. Well, at least we will have to solve the problem of sample selection from ever-growing libraries that now reach gigabytes, and we will have to do something about the rigidity of recorded samples once they are selected. There is certainly room for more research here. As storage limits disappear, the real limits of sampling are becoming apparent, and old solutions such as work from my lab on Spectral Interpolation Synthesis (Combining Instrument and Performance Models for High-Quality Music Synthesis, 1998) and other work on physical models are re-emerging.

Composing in the Future

Recently, there has been a resurgence of work on automated computer music composition. Every innovation in Artificial intelligence – rule-based expert systems, constraint systems, production systems, Bayesian approaches, neural networks and now various kinds of deep learning – has been applied to model the compositional process. We can expect this trend to continue.

In my view, recent work, while technically impressive, has been musically disappointing. Perhaps the success of deep networks in other areas has misled researchers into putting too much faith in data-driven learning methods. Composition is regarded by many as a problem of imitation: Train a machine learning algorithm with examples of music and try to generate something similar. But how many composers aim to (merely) imitate? Composers have not played a large role in recent research, and in many ways, earlier research by composers produced more musical results. Composers have a better understanding of what composition is really about, and it seems that deep learning is no substitute (yet) for human understanding. Then again, with another 10 years’ growth in computational power and the qualitative changes it will bring, maybe time will show that I am just taking a short-sighted view.

There is clearly room for more research here, and in the long run, we will see a slow and steady progression beginning with simpler tasks such as making drum loops, harmonization and creating musical textures. From there, perhaps we will develop composition systems that work with in highly constrained settings: improvising over a set meter and chord progression, composing percussion tracks or bass lines given a set of parts, or generating call-and-response melodic units. Eventually, we will come to understand higher-level structures, music anticipation and surprise, and music design to the point we begin to see truly original musical creations by computer.

Performance in the Future

Live performance with computers is still nascent. There are some stunning pieces in the repertoire, and plenty of techniques from composed improvisation to computer accompaniment, but let us be honest and critical here. Interactive systems are largely based on triggers to step through fixed sequences, simple responses to simple input patterns, or just random but interesting choices. Machines have little understanding of tempo, timbre, form, anticipation or surprise, and it is as much a stretch to call computers true collaborators in 2020 as it would be to call the pianoforte a musical collaborator in 1750.

Computer accompaniment systems coordinate with musicians at a finer time scale by tracking performances note-by-note. Work with Gus Xia shows that deeper musical understanding can dramatically improve prediction in collaborative performance. (Xia, Wang, Dannenberg, & Gordon, 2015) So far, computer accompaniment systems are quite shallow and fail to adapt as collaborators might. These systems are also brittle, typically applying only one method of listening or processing input, whereas musicians have a much richer repertoire of techniques including score analysis, phrase analysis, entrainment to beats, leading and following, giving and accepting visual cues, source separation, and exceptional musical awareness.

One of my research directions is to enable human-computer collaboration in the performance of beat-based music, an area largely ignored by Computer Music research. It is not clear who would actually perform with such systems, but it is an interesting challenge. In any case, we have a long way to go to develop more computational music understanding for live collaborative music performance.

A "Moonshot Project" for Computer Music

My colleague Rowland Chen created an interesting challenge that I believe exemplifies the current problems in Computer Music research. (Chen, Dannenberg, Raj, & Singh, 2020) Just as the goal of putting a man on the moon stimulated an array of technical advances in space exploration, with wide-ranging and important spin-offs, I believe a good "Moonshot" project might stimulate and stretch Computer Music research.

Jerry Garcia was a founding member of the Grateful Dead. He is dead, but millions of fans miss him, and thousands of hours of live recordings survive. What if we could create a faithful imitation of Jerry Garcia? The problems we would have to solve span the range of Computer Music concerns, including:

- Sound: Model Garcia's electric guitar sounds, including effects, amplifier distortion and sound propagation. Vocal sounds seem even more difficult.
- Control: Isolated guitar sounds are not enough. Perceived sound is influenced by articulation, bends, vibrato, frets and fingerings, all of which are time-varying, constrained by physics, the neuro-musculature system and mutual dependencies. Again, the singing voice is yet more difficult.
- Composition: The Grateful Dead are known for long improvisations and launching the "jam band" movement. One would expect a "Jerry Gracia" model to create new improvisations with long-term coherence, interaction and collaboration with human bandmates and faithful adherence to style. (Perhaps a continuing evolution of style is also necessary to keep fans interested and to justify new performances.)
- Collaboration: Part of the essence of the Grateful Dead is the collaboration among the band members in constructing extended "jams." Musical coordination exists at all levels from beat- and measure-level synchronization to larger sections and transitions.
- In order to accomplish all this, it seems necessary to greatly extend the state-of-the-art in machine listening, especially source separation techniques. If we could isolate instruments in the 10,000 hours of Grateful Dead concert recordings that are available for study, we would at least have a wealth of interesting data. Even with that data, we need advances in the analysis of structure and style in those performances.

Whether we actually embark on a "moonshot" project, it is a good practice to set goals and to dream big. In my experience, real objective musical goals are invaluable in setting the research agenda.

Conclusions

If we stand back far enough, we can see Computer Music as a grand undertaking to understand and automate *every* aspect of music making, with a clear progression:

- From primitive sound generation and reproduction, we have learned to create new sounds. Research continues to explore new sounds as well as to create better models for known acoustic sounds in all their richness and complexity.
- Beginning with simple event lists and other score-like representations, we have developed more complex and dynamic control approaches, leading to imitative computer-generated compositions and to interactive, responsive music systems.
- From early performances with fixed media, we have developed computer accompaniment systems, responsive robot musicians, and we have begun to study collaborative music making in greater generality.

I believe these trends help us to anticipate what the future will bring: Richer sounds and better synthesis models, better understanding for building higher-level musical forms from phrases to entire music compositions, and more sophisticated approaches to collaborative music making between humans and machines.

While these themes seem to be predictable, the exponential growth of computing power makes the details hard to even imagine, and we should expect *qualitative* changes on par with the shift from mainframes to laptops or books to Internet. These changes will continue to surprise us, but they will also open new and interesting avenues to pursue our goals.

Ultimately, our attraction to modeling, automation and computation in music is driven by the natural human urge to explore and learn. Let us hope that through this experience of constructing knowledge, we also learn to use it wisely for the benefit and enjoyment of society.

Acknowledgements

I wish to thank the KEAMS organizers for the occasion to organize these thoughts. My work would not be possible without the support the School of Computer Science at

Carnegie Mellon University and my many stellar colleagues and students, from whom I have learned so much. Throughout this paper, I have referenced particular papers I had in mind, but I also mention areas full of great contributions by many additional researchers. I believe a quick Internet search with obvious keywords will lead you to their papers, and I hope dozens of authors will forgive me for omitting references to their work here. I have certainly learned a lot from my Computer Music colleagues, whose friendship over the years continues to make this a great journey.

References

Anderson, D. P., & Kuivila, R. (1990). A System for Computer Music Performance. *ACM Transactions on Computer Systems, 8*(1), 56-82.

Bilbao, S. (2009). *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics.* Wiley.

Block, G., Chabot, X., & Dannenberg, R. (1989). Jimmy Durante Boulevard. On *Current Directions in Computer Music Research (CD-ROM)*. Cambridge, MA: MIT Press.

Chen, R., Dannenberg, R., Raj, B., & Singh, R. (2020). Artificial Creative Intelligence: Breaking the Imitation Barrier. *Proceedings of the 11th International Conference on Computational Creativity* (pp. 319-325). Association for Computational Creativity.

Collinge, D. J. (1985). MOXIE: A Language for Computer Music Performance. *Proceedings of the International Computer Music Conference 1984* (pp. 217-220). San Francisco: International Computer Music Association.

Dannenberg, R. (1985). An On-Line Algorithm for Real-Time Accompaniment. *Proceedings of the 1984 International Computer Music Conference* (pp. 193-198). San Francisco: Computer Music Association.

Dannenberg, R. (1986). The CMU MIDI Toolkit. *Proceedings of the 1986 International Computer Music Conference* (pp. 53-56). San Francisco: International Computer Music Association.

Dannenberg, R., & Derenyi, I. (1998). Combining Instrument and Performance Models for High-Quality Music Synthesis. *Journal of New Music Research, 27*(3), 211-238.

Dannenberg, R., & Mercer, C. (1992). Real-Time Software Synthesis on Superscalar Architectures. *Proceedings of the 1992 International Computer Music Conference* (pp. 174-177). International Computer Music Association.

Dannenberg, R., Pellerin, H., & Derenyi, I. (1998). A Study of Trumpet Envelopes. *Proceedings of the International Computer Music Conference* (pp. 57-61). San Francisco: International Computer Music Association.

Dannenberg, R., Sanchez, M., Joseph, A., Capell, P., Joseph, R., & Saul, R. (1990). A Computer-Based Multi-Media Tutor for Beginning Piano Students. *Interface - Journal of New Music Research, 19*(2-3), 155-173.

Eigenfeldt, A. (2007). Real-time Composition or Computer Improvisation? A composer’s search for intelligent tools in interactive computer music. *Proceedings of the Electronic Music Studies 2007.* www.ems-network.org/IMG/pdf_EigenfeldtEMS07.pdf.

Lewis, G. (2000). Too Many Notes: Computers, Complexity and Culture in "Voyager". *Leonardo Music Journal*, 33-39.

Mathews, M. (1969). *The Technology of Computer Music.* MIT Press.

Mathews, M. V. (1980). The Sequential Drum. *Computer Music Journal*, 45-59.

Mazzoni, D., & Dannenberg, R. (2002, Summer). A Fast Data Structure for Disk-Based Audio Recording. *Computer Music Journal, 26*(2), 62-76.

Rowe, R. (1992). *Interactive Music Systems: Machine Listening and Composing.* Cambridge, MA: MIT Press.

Vercoe, B. (1985). The Synthetic Performer in the Context of Live Performance. *Proceedings of the 1984 Interntional Computer Music Conference* (pp. 199-200). International Computer Music Association.

Wessel, D., Lavoie, P., Boynton, L., & Orlarey, Y. (1987). MIDI-LISP: A LISP-Based Programming Environment for MIDI on the Macintosh. *Audio Engineering Society Conference: 5th International Conference: Music and Digital Technology.* AES.

Winkler, T. (1998). *Composing Interactive Music: Techniques and Ideas Using Max.* Cambridge, MA: MIT Press.

Xia, G., Kawai, M., Matsuki, K., Fu, M., Cosentino, S., Trovato, G., . . . Takanishi, A. (2016). Expressive Humanoid Robot for Automatic Accompaniment. *SMC 2016 - 13th Sound and Music Computing Conference, Proceedings*, (pp. 506-511).

Xia, G., Wang, Y., Dannenberg, R., & Gordon, G. (2015). Spectral Learning for Expressive Interactive Ensemble Performance. *Proceedings of the 16th International Society for Music Information Retrieval Conference*, (pp. 816-822).