Music at MIT Oral History Project

Barry Vercoe

Interviewed

by

Forrest Larson

***with* Christopher Ariza**

April 24, 2012

Interview no. 2

Massachusetts Institute of Technology
Lewis Music Library

Transcribed by MIT Academic Media Services and 3Play Media.
Cambridge, MA

Transcript Proof Reader: Lois Beattie, Jennifer Peterson
Transcript Editor: Forrest Larson

Table of Contents

Contributors

Barry Vercoe (b. 1937) is Professor Emeritus of Media Arts and Sciences at MIT. Since 1985 he has been at the MIT Media Lab. And from 1971 to 1985, was Professor of Music at MIT in the Department of Humanities. He is among the first generation of engineers in the field of computer music. His achievements include creation of the widely influential Csound software and development of technology for real-time interactivity between live performers and computers. He is also a composer of orchestral, choral, and chamber music as well as notable works combining computer generated sound and live acoustic instruments.

Christopher Ariza was Visiting Assistant Professor of Music at MIT, 19XX–19XX. He is a composer, software developer of musical tools for live electronics in algorithmic generative systems.

Forrest Larson, Library Assistant at the Lewis Music Library, has attended training workshops in oral history methodology and practice at Simmons College and by the Society of American Archivists, and is a member of the Oral History Association. He is also an active composer and violist.

Interview conducted by Forrest Larson on April 23, 2012, in the MIT Academic Media Production Services studio. Second of two interviews. First interview August 19, 2011. Duration of the audio recording is 1:54:17.

Music at MIT Oral History Project

The Lewis Music Library's *Music at MIT Oral History Project* was established in 1999 to document the history of music at MIT. For over 100 years, music has been a vibrant part of the culture at the Massachusetts Institute of Technology. This history covers a wide variety of genres, including orchestral, chamber, and choral musical groups, as well as jazz, musical theater, popular and world music. Establishment of a formal music program in 1947 met the growing needs for professional leadership in many of the performing groups. Shortly thereafter, an academic course curriculum within the Division of Humanities was created. Over the years, the music faculty and alumni have included many distinguished performers, composers, and scholars.

Through in-depth recorded audio interviews with current and retired MIT music faculty, staff, former students, and visiting artists, the *Music at MIT Oral History Project* is preserving this valuable legacy for the historical record. These individuals provide a wealth of information about MIT. Furthermore, their professional lives and activities are often historically important to the world at large. Audio recordings of all interviews are available in the MIT Lewis Music Library.

1. Family and musical (00:15)

LARSON: It is my honor and privilege to welcome Barry Vercoe, Professor Emeritus of Media Arts and Sciences at MIT. From 1971 to 1985, he was Professor of Music at MIT in the Department of Humanities. And since 1985, he has been with the Media Lab. Thanks again for coming for the second interview.

Also, assisting me is Christopher Ariza. He is Visiting Assistant Professor of Music at MIT. He is a composer, software developer of musical tools for live electronics in algorithmic generative systems. And thank you, Chris, for coming as well.

So, Barry, I wanted to ask you about some stuff from the early years of—of computer music, more, kind of, aesthetic, kind of, questions. The early days—computer music research—a lot of it—I mean, modernist composers were very interested in that. But there were some other things, other interests, of some of the—the founding figures of computer music, like Max Mathews (1926–2011).

There were other, kind of, pragmatic goals. Can you talk about that? Then we'll talk about some of the—the more, kind of, modernist stuff. But I'm curious about what was driving some of the original research.

VERCOE: Well, it really has to do with control, that composers wanted precise control over the sounds they were going to get. And in the early days, when I first was in electronic music with analog synthesizers, I worked a little bit with Mario Davidovsky when he showed up at MIT—not MIT, at University of Michigan, as a guest composer there in, probably, 1963.

The prospect of moving from cut-and-splice [analog audio tape] techniques, of course, but—for which he was the—the master, into the development of hardware such as the Moog synthesizer and things, would put you into a situation where it was certainly real time, but you would be sitting on stage there, tuning these analog synths—[laughs] analog modules and, sort of, having to worry about the tuning. Because they would float around.

Whereas with computer systems, you have precise control over the—the pitch of things and—and the—the pacing of events, and so forth. And that's what appealed to me. I had come to this, I suppose, as a composer since I had studied with Ross [Lee] Finney [at the University of Michigan] when I first came to this country. Ross—I may have told you before—had been a student of Alban Berg, and he of Schoenberg, of course. So I was into this thing of some kind of control when I just was writing music for instruments.

And so that idea, that feeling sort of came across, I—my earliest pieces, when I—when I first came here were 12-tone pieces. And when I got into computer music, those pieces also were 12-tone pieces. My clarinet piece [*Setropy*, for clarinet and piano], for instance, is strictly 12-tone.

I tend to come up with tone rows that were harmonics, or more of the Berg-ian rather than the [Anton] Webern tradition. I'm still believing in—very much in harmony. I'm still very much in love with choral music, which is usually fairly tonal, not always. So the element of control of the medium was something that took me in the direction of the—the computer as the instrument.

LARSON: Some of the other people seem like there were—there was an interest in, kind of, just seeing, kind of, what the computer could do with sound. And, I mean, there was some early stuff where they were creating, you know, recreations of—you know, tonal music, even some pop tunes, and stuff like that. So it wasn't necessarily musically innovative, but it was technically innovative. Can you talk about what were some of the reasons that people were doing that, and, kind of, what their pragmatic goals were?

VERCOE: Well, the initial fascination of having a musical performance without having to worry about performers would be one of the things. That you could have something happening and just be in control, once again, of it yourself. That goes so far.

Because ultimately, when you want to get something that emulates the capacities of an ensemble on one or a number of computers that are working in collaboration, you get into some fairly heavy technical stuff, communication between different operating systems and so forth, all running in real time with a precision that you really want to be very, very accurate. And it ends up, as I learned when I did a live performance of my *Synapse* for viola [and computer sound], without running from reel-to-reel tape, but just having the compu—one computer follow the—Marcus Thompson, the violist, and another computer doing the synthesis of the accompanying part, in sync with the live viola, where the live viola did have, now, an element of autonomy over the—the—whatever was going on.

I began to feel very nervous, not about live performers, but about the performance of the computers, that they would screw up in some way or other. So control goes only so far. When you have so much complexity going on, then it becomes almost, sort of, a big neural system. And you then worry about the behavior of that neural system.

ARIZA: I think an interesting aspect of this opportunity for control is that the control could be applied to other musical parameters of pitch. The serialists, of course, were interested foremost in organizing pitch structures.

VERCOE: Mm-hm.

ARIZA: And it seemed to me that a lot of early computer music practitioners sort of reveled in the opportunity to control other musical parameters, other than pitch. I'm wondering if you have any thoughts about that.

VERCOE: Well, the earliest computer music that came out of, say, Illinois [University with the Illiac string quartet ["Illiac Suite" (1956), composed by Lejaren Hiller and Leonard Issacson, using a computer program to create a score to be played by conventional string quartet] and so forth was not actually fo—done for pitches at all. That—that is to say, com—computer-generated pitches. It was just something, a score—writing a score—was then to be played by live performers.

And that continued, in a way, to be the tradition in Europe, that you get the works of Gottfried Michael Koenig, and so forth. And they were concerned about using the computer as a way of composing, and those sco—those pieces were then played by live musicians. The thing that was happening in America, and probably because it had a sort of a technical advantage, was the composers then tended to focus on the computer as the realizing of—of the sounds themselves. And at some—in some cases, the sounds might in fact—well the sequence of pitches may be in fact computer generated, mostly the earlier composers were doing the composing themselves. And having the computer, of course, then just generate the sounds as a substitute for live performance.

ARIZA: Mm.

LARSON: So, in the early days when you weren't—you didn't have a real-time control [of computer processing for synthesis] with the—the computer, there's a lot of—of significant preconceptualization that has to—to happen. And even before the—you know, the era of—of computer music, there's lots of composers who really valued a lot of, kind of, preconceptual thought.

VERCOE: Mm-hm. Mm-hm.

LARSON: And as [with] computers—we have this real-time control. I'm wondering if you have any thoughts about what—is there something that might be—might have been lost, in terms of a kind of artistic discipline that comes with the preconceptualization, where you don't get, you know, real-time, kind of, feedback? In the same way that if you write a big orchestral piece, you don't get real-time feedback until you hear the—the performance. And I'm wondering if there's a similar kind of thing that with—with computers and something that—that might be—might be lost?

VERCOE: Yes. Yeah, well, I should—we should append to the word "control" the idea of real time or not real time. So, there is control that can be there in the form of how you create the score, what kinds of parameters are in the score, that you then depend on technology to realize accurately. But until about 1990, none of that was in re—was happening in real time. So that the idea of going into a studio for thirty-minutes and coming out with a twenty-minute piece, which was the case—was a possibility a few years later when you have, sort of, automation to the point that composers are giving up control almost entirely, is—is not the situation in—in the early days, where composers were themselves responsible for the score, and the computer was just simply realizing the sound.

But that was putting the—the sound realization onto some storage medium, typically, you know, a disk initially, but then coming out through an analog-to—digital-to-analog converter and recording the sound on analog tape, typically reel-to-reel. And that was where you then got into something that we—you could hear back in real time. It might have taken two hours to synthesize something that took two minutes to then play back. So you have a 60:1 computer-time to real-time ratio, which was pretty common in the early days. So, there wasn't the sense of real-time control happening in those circumstances. Wasn't until—in fact, the first real-time performance was with my Csound that I took—a version of that, that I took to a computer music conference in Glasgow in 1990.

And that's the first example we have of something happening in real time, the synthesis happening in real time, where you could now for the first time have interaction with it. And that was—that was a huge change, the—perhaps the biggest change in computer music to have real-time interaction so you could—you could have the computer listening to you in real time and vice versa. You could be listening to it in real time and responding. So suddenly you—into a situation there where it's rather like live performers listening to one another. And that makes the whole composition and creation thing go into a very different phase.

LARSON: Right. I mean, there—there's real time performative aspect, but the—the real-time, kind of, compositional aspect.

VERCOE: Yes.

LARSON: And do you think that there's anything possibly lost in terms of the—the kind of discipline that it took when there wasn't a real-time feedback with hearing sounds that you're—you're synthesizing? I mean, we all see that now—that real time is a—is a positive thing because you want to hear it. But I've just wo—I've just wondered if—if something might have been lost in terms of a particular kind of artistic discipline required—

VERCOE: Yes. I—I know what you're driving at. Most certainly, if you're forced to wait and wait for something, you're going to think about it more carefully, particularly if you're going to have to in—invest four hours of computer time to hear one—four minutes of music, or something. You're going to take a lot more trouble to notate precisely what you want. And that is a—puts one in a reflective mode, and when you're composing, and you, sort of, think about something, and go back and revise, and revise, et cetera. And naturally that's going to benefit the piece most of the time.

Sometimes it pays to be very natural and, sort of—just being able to do—do things quickly. But being able to think about a piece and refl—reflect on it is usually a good thing. And so when com—the computers were very slow, there was a lot of thinking that was going on. The computers were thinking, too. But the humans had a lot of thinking time, as well.

And I think you're right, that that did benefit the early pieces. Although the early pieces were not really very exploratory. They were not very innovative. They tended to—to do—uh—simple things. They were sort of not getting into the big, sort of, massive amounts of sound that the analog studio people—the Mario Davidovsky pieces, for instance, were able to get into in the—in that—I'm talking here about the early '70s. And so it took a while for computers to get to the point that they could actually match the sound banks that you got from the studio.

LARSON: Right. Right. In those early days—I'm probably thinking more so of the—the analog, there was this—this—this concept of what they called organized sound. And in some ways there was this notion that we had a new—almost a new kind of art form that was using sound. But it wasn't based upon, you know, traditional parameters of pitch and rhythm, but the timbre, texture, you know, and—and color.

VERCOE: Yeah, well of course you had—you had that era in Paris, in the early Paris days of—

LARSON: Right.

VERCOE: —found sounds, and so forth. [Ed. Note: Musique concrete began in the early 1940s.]

LARSON: Right, right.

VERCOE: Or the—

LARSON: Right

VERCOE: —work [electronic music] of [Vladimir] Ussachevsky [1911–1990], or whatever.

LARSON: Right.

VERCOE: You had a lot of that in the analog domains. And you could have that same thing, of course, showing up, then, in computer music, once you got to the point that you could have the computer modify the sounds that you were working with, go through and analyze the—the special content, and shift it around in some way or other. And once those techniques became fairly advanced, that of course was a fun way—fun thing to do, to take sounds and pull it apart using digital techniques, and put it together in other ways. And so there the computer had another perspective on sound that the analog studios didn't.

LARSON: Mm-hm. Had you done much work, kind of, with the—with those kind of concepts, either with other composers—you know, in some of the workshops you—you did with composers or your—your own work, with this—this, kind of, organized sound, kind of—principle?

VERCOE: Myself, not really. I—I suppose in my *Synapse* piece, there was an example there where I took an analysis of—of a string—violin sound and took it apart and re-synthesized it and so that I could play it again in—at—in different pitches without having the spectral change. And that's the essence of what you could do when you've got phase vocoders and things later on. You could actually make pitch and time shifts without, sort of, having the munchkin effects of speeding up and slowing down. But that took the development of phase vocoders, which was mid '70s, late '70s, before those things became realistic.

ARIZA: As far as the opportunity to manipulate sound files in the trajectory of MUSIC 11, MUSIC 360, Csound, where did these opportunities start to become available?

VERCOE: With MUSIC 360, they were not really available much at all.

ARIZA: Uh-huh.

2. MIT Experimental Music Studio and Founding of the MIT Media Lab (16:24)

VERCOE: With MUSIC 11, once we had the computer to ourselves, in the—the early MIT Experimental Music Studio, we could almost get our hands on things. [laughs] And that's when you began to find the awakening of the possibilities of being able to do something, have—have a greater level of control over the digital processed sound. But the real control didn't come until 1990 with this—the first real-time Csound performance, which was at the ICMC [International Conference on Computer Music] meeting in Glasgow, 1990.

LARSON: So you were one of the founding members of the—the MIT Media Lab. But prior to that you had the Experimental Music Studio, which started in 1973. And then that moved to the Media Lab. As a—as a founding member, what was your understanding of the—the mission of the Media Lab? You know, the MIT Council for the Arts [Council for the Arts at MIT] wasn't very involved in—in—in getting that—that started. But what was your understanding and vision of the—the Media Lab?

VERCOE: Okay. Well, for me, it was to maintain the kind of autonomy that I originally had when I had my own lab, which I had with the EMS [Experimental Music Studio] studio. That I had taken—I had got by taking over Amar Bose's lab in—when I first came here in 1971. Bose was just moving out of MIT, ceasing to do research here, though he still continued to teach. And—but I then had a—a lab with a computer, and it was all to myself. And I think a lot of the things we did through the composer workshops that you referred to, lots of composers coming in, lots of concerts in Kresge [Auditorum], and so forth, was something that just came out of that one institution, the Experimental Music Studio.

And we had total control over the kind of music we did, or at least the kind of music we put on for the concerts, and so forth. Much of it was interacting with live instruments. That's the tape and instrument sort of thing, reel-to-reel recorded parts. In moving to the Media Lab there was sort of a pooling of missions and things like that. Initially, we thought, as a group of six people—Marvin Minsky [MIT Professor of Computer Science], Seymour Papert [MIT mathematician and computer scientist]—there were six of us there, Nicholas [Negroponte, Chairman Emeritus of MIT Media Lab] and so forth—that we were going to learn, perhaps, to work together, and that we had all sorts of wonderful things would come out of collaborations.

The collaborations didn't really occur with—between the faculty. We were—each having come from our own bailiwick of some kind, the first thing was to move that stuff into this new building, and then try to get it running just the way it had been running previously. And that sort of was an accomplishment. The idea of working together—I had a first thought, "Well, I've never made a film. Wouldn't it be fun to make a film with [Richard] Ricky Leacock [documentary film director] or something, or get his help?" I never did succeed in making a film. [laughs] I just focused on the music part of it.

And so, initially, there was, sort of, six faculty members there, each with their own individual missions. The over—overriding mission of the Media Lab, of course, was to encourage creative exploration, and so forth. That was a reality. But that was—so that was fun.

So, I was no longer having to write lots of proposals to the National Science Foundation. I could write them instead to the National Endowment for the Arts. So I had help with—from Judy Whipple [Administrative Assistant at the Media Lab] doing that.

And, so we could embark on a course of music creativity and innovation there, with the under—the underpinning of the—the technical innovations that we had also going, through the work of the graduate students. That able—enabled us to do new, wonderful things. But at the same time, it was all in this controlling environment by people at the Media Lab, who didn't always appreciate the arts.

It was, you know—it was intended to be something that was driven by creativity. But the arts in their finer form were not always appreciated by the engineers. And one can understand that.

Same thing happened at IRCAM in Paris. The—the set—set-up there that [Pierre] Boulez was the—Institution for Coordination of—Research and Coordination of Acoustics and Music was the goal. But there were people who had come in as composers—[Karlheinz] Stockhausen was there—they were the—the leaders, the gods. And the engineers simply served them. They did their own research, but when the composers were wanting to do things, the—the engineers would be at their beck and call.

There wasn't a real collaboration. There weren't people there who really did both, ex—with one exception, and that was Jean-Claude Risset. He was the—the first, sort of, director of computing, you might say, and scientific director. And he found the whole thing uncomfortable. And he moved down to Marseilles within about five years.

LARSON: And you had him here as a composer in residence later on—

VERCOE: Oh, yes. Yes, yes.

LARSON: —which I have a question later. So this idea of the Media Lab, the word "media," that obviously encompasses a lot of things. And it—a lot of the work going on there now is very much, kind of, engineering based and—and less, kind of, arts based.

VERCOE: Mm-hm. Mm-hm.

LARSON: Was that—is that a drift from some of the original intentions?

VERCOE: [sigh] I suppose. I think we were all being very optimistic about the extent to which we could maintain artistic integrity and not be having to pay too much attention to technical outcomes. But, of course, the sponsoring members of the Media Lab would be looking for the fallout that would come from artists experimenting and generating innovations, and so forth. They were looking for the innovations that they could pick up as things that they could take into industry, products or something, where they could get a royalty-free, non-exclusive license to commercialize. And, of course, those things were more technical, rather than artistically innovative.

3. Graduate Degree Program at the Media Lab (23:10)

LARSON: Mm-hm. So when the Experimental Music Studio moved to the Media Lab, were there still undergraduate computer music courses? Or was it—did it become, kind of, a graduate kind of program?

VERCOE: As far as the course is concerned, I usually ran it at two levels—there were people who could register as graduate students, take it for graduate credit, and people who would register as undergraduates taking it for undergraduate credit. It was usually the same lectures. But the graduate students would be expected to do more. Maybe there'd be other assignments, or something, for the graduate students.

But insofar as that any graduate student probably didn't have a background in computer music before coming, they were both at the same level in many ways. Young bright undergraduates are just as capable as the grad students, as you're probably aware. And so they were, sort of, all of a—all of a one, as it were, with just a little bit more push for the graduate student to do something perhaps more innovative or more technical. Because they were, after all, going for a master's degree or Ph.D.

LARSON: Was there a period of time where there wasn't an undergraduate, you know, opportunity to study computer music there? Is that one of the reasons why the Music Section started an undergraduate computer music lab? I'm just wondering, was that filling a need for that? Or was that just a separate thing that the Music Section—

VERCOE: Well, it came—I—I understand what you're asking. And it came from the fact that in the Media Lab we had established this degree program. And a degree program based on doing some research and innovative things that would—had to have equality amongst all the disciplines that were operating at the time.

And so we were wanting to make sure that people who came out of the Media Lab had a degree that would mean something, and so that people could go and seek a job. I had some very good students who came through and did a master's with me, with the idea, ultimately, of doing a Ph.D. so that they could become a university teacher. But in the beginning, it wasn't commonly regarded to have a degree—a Ph.D. from MIT as a good qualification to get a university job somewhere. So I lost some of my best students to go—who'd go off then and do a Ph.D. at UC Berkeley or some other place, where the university itself and the—the programs there already were—had a qualification associated with them.

Later on, that idea that a degree from MIT Media Lab, where the focus there was on, I suppose, interdisciplinary, cross—cross-disciplinary ideas, later on they became very advantageous for the students. So that when the students first arrived, of course—arrived in the early days in the Media Lab, with six of us faculty, and no two of whom were in the same field, and we—and we tried to work with—together, as I said earlier. But we didn't really. It was the students who then found themselves in this highly interdisciplinary environment and soon learned the art of lateral thinking.

And whereas in the early days in the Media Lab, that was not very valuable outside of the Media Lab, not highly regarded. But later on, as academia moved, I suppose, around the country, the idea of being an expert in—in lateral thinking, about—through dif—across very dif—different disciplines was an asset. And very soon, after about eight or ten years, we found that the—the hitherto narrowly defined departments were then looking to—to hire our students, because they suddenly had this cross-lateral thinking as—as an experience. And that was a really—a big accomplishment of the Media Lab. That was one of things that Nicholas Negroponte, the founder of—of the Media Lab, thought would happen, and thought would happen sooner. It took a little while, but it did happen.

LARSON: So in a previous interview we talked about the—the summer composer workshops that you had done and touched on some of the composer in residencies that you had. That was a pretty exciting time, and then that—that died out. What were some of the reasons for that? And it's—it's certainly something that I have—have missed.

VERCOE: I got tired in the end, [laughs] running these workshops every summer. I would be run off my feet for a—a period of six weeks, with very little sleep and getting these pieces—or getting the incoming students all up to a certain level. And of the 30 students who'd come into the workshops, I would then choose about 10 who were, sort of, commissioned to write pieces for a concert in Kresge, which was already scheduled. We just didn't have any music yet.

And, so that was a very tiring period, each summer to be doing that with no, sort of, break, then normal teaching during the year. So that later on when I then do—did things like went—went off to Paris and work at IRCAM for myself for a little bit and then come back in the Media Lab—and we were then beginning to get funded by the National Endowment for the Arts. Then that put us in a, sort of, a slightly different mode, where the composers were coming in with commissions.

Some of them would come in and stay until they had finished their work. Some of them would come in, get the feel of the place, go away and compose, and then come back and realize the piece, but always with the assistance of students as technical assistants. In that sense, those students were then functioning like the engineers at IRCAM, who would be assisting the composer to realize their—their music.

But this then made it—made it perhaps unnecessary, or perhaps—I'll say unnecessary for us to be running summer workshops when we were bringing composers in from outside. These composers now were coming in with grants. But there were fewer of them. There'd be two or three composers a year who would come in for a few months at a time and do a piece. And that meant that the output came from those commissions, rather than from the summer workshops, where people were, in effect, paying us to write their pieces. [laughs]

LARSON: Yeah. I mean, there was a period of time when MIT was really on the forefront of—of computer music. And—and I came to MIT in 1985 and went to many of those—those concerts. And that's something that—that I've missed. And I—is there any possibility that—that maybe in the future that there may be some?

VERCOE: Oh, yes. Well, I also in '85, I was at the time in Paris working with Larry Beauregard, the flute player that was in the—you've seen in some of my videos of interactive direction—that's the beginnings of the real-time interaction. We have a flute being tracked by the computer. And the—the computer part, computer accompaniment, being synced to the live player.

Larry was a—a wonderful performer. As a flute player, he had graduated from, I guess, McGill University and went—or maybe it was University of Toronto. But he was Canadian. And then went—he got a job straight out of school as the principal flutist in the Boulez ensemble. And he had sort of reached that apex, in a way, of his career and was looking for something different, and was intending to come to MIT and to be in the—in the Ph.D. program, and be part of a small ensemble of performances, and so forth.

Sadly, Larry then became very ill and died of colon cancer shortly after those videos that I made. And I was very disappointed, because I was looking for him to be the—the nexus of a small performance ensemble that we would have at the Media Lab. Just as many schools have resident string quartets or something, we'd have perhaps a resident ensemble of—Larry would be the—the ringleader, and we'd pull in other players.

And—and we'd have, then, a string of pieces that would be written for those particular players. And the students then would become involved in figuring out what it is that is so special about live performance and how computers can, sort of, synchronize with live performers, and so forth. But Larry was my initial key to that. And his passing meant that that didn't happen. And I was bitterly disappointed at that.

LARSON: Mm-hm. So I wanted to ask you about Professor Paul Earls, who had been—was Assistant Professor of Music at MIT and also a fellow at the MIT Center for Advanced Visual Studies.

VERCOE: Yes, yes.

LARSON: How much did you know his work? He did both analog and, you know, digital electronic music, and—

LARSON: Yes, yes, yeah. Paul was from Duke University. He had been active there and then came out to work here in CAVS [Center for Advanced Visual Studies]. I liked his work a lot. It was difficult at the time to integrate CAVS and that part of the—of the Architecture Department with the Media Lab.

Nicholas [Negroponte] tended to run the Media Lab as an independent organization and not automatically embrace a lot of the other smaller—smaller initiatives or other initiatives around MIT to make it one big, happy place. So it was difficult for me to bring Paul Earls into—into the—into the Media Lab, just as it was difficult for me to bring people in the Music Department who were studying film music or something like that into the Media Lab as—as an example of academics looking at the evolution of media. There were specialists in the—in the Music Department who were focusing on things like that, and still are.

LARSON: Right.

VERCOE: But bringing them into the Media Lab where the—the focus to—for us to survive financially had to be on producing, I suppose, scientific results, and—meant that we couldn't be as encomp—all-encompassing as we'd like to be.

LARSON: Mm-hm. Did you happen to see the performance of his piece called "Icarus"? It's a thing called a "sky opera." Did you—?

VERCOE: I did see parts of it. I don't recall a lot of detail, I have to say.

LARSON: Uh-huh.

VERCOE: Yeah.

LARSON: Uh-huh. And he did some really interesting things with music and lasers—

VERCOE: Yes, yes.

LARSON: —some interesting sound, kind of, installations.

4. Notable composer residencies. Innovations. EMS 25th Anniversary (30:41)

ARIZA: I guess on that note, it's—it's interesting to think about a—a role in the form of computer music that doesn't use live performers has often tried to provide additional visual elements. We have a lot of electro-acoustic music that involves video. And there's been a lot of attempts to—to bring other media in to fill the gap when a performer is not there. I'm wondering—I know that you've—you really appreciate the opportunities to work with performers, but I wonder, in terms of computer music that stands alone, what you think about this missing visual element.

VERCOE: I love some of the music that does—is just speakers alone. We did a lot of pieces like that, in fact some of them big, four-channel pieces where the missing element were—was perhaps the lack of human performers, and everybody focused on looking at a small group on stage, which was replaced by six speakers all around Kresge Auditorium where you were just involved in this whole, three-dimensional, almost—because there were some of the speakers at the back were sort of coming above you. And, for instance, one of the pieces from—called "In Winter Shine" that's on the CD from that—was just a wonderful example, where the composer there was doing some wonderful things about meshing sounds together, having sounds or sound banks that were related to one another. [Ed. Note: from *Digital rewind [sound recording] : MIT Experimental Music Studio : 25th anniversary, 1974-1999*.]

This is [James] Jim Dashow's piece. And Jim was a wonderful composer. He came out of Brandeis [University], actually, went to Rome and is still in the outskirts of Rome at this—to this day.

And his way of making some continuity from the—these—this almost three-dimensional sound and big sound banks of, sort of, essentially inharmonic tones, was to mesh these so that when you—when there were two sounds that were contiguous—two sound banks contiguous, he would have some common tones between the two of them. So you'd hear this one sound, you'd hear another sound, and it would seem to relate. And you'd sort of wonder why, because there was no seemingly harmonic relationship.

But it was rather like the [Franz] Schubert—common tones between—in Schubert modulations. There would be some common tone between the two key centers. And Schubert would, sort of, play on that. And Jim was doing that sort of thing as a common tone between components of big banks of sound. And it made the things have just a great continuity.

And—but that came from a very determined use of the computer, where he could rely on things being exactly so, and then formed his own relationship between the—the sound clusters through that technical means. You wouldn't know that in hearing the piece. You would just feel that, "Gosh, this seems to be consistent. There's a coherence here. Wonder what it is!" And that was it.

LARSON: So backtracking a little bit. Jean-Claude Risset [b. 1938] had done a composer in residency here. And he wrote an interesting piece called "Duet for one Pianist," where the second piano is—is computer controlled. And that's an interesting aspect of computer music, too. And give me thoughts about that. And also, I wanted to ask you about this thing that he put together called *Sound Catalogue of Computer Synthesized Sounds.* It's kind of a—a cataloging documentation of— [Ed. Note: *An Introductory Catalogue of Computer Synthesized Sounds.* 1969.]

VERCOE: Mm-hm. Well, that sound catalog predated the—the Media Lab, even, and predated the—the stuff at MIT.

LARSON: Right. That was from 1969, yeah.

VERCOE: That was—was when Jean-Claude was at Bell Labs [Bell Telephone Laboratories, Acoustics Research Division. Computer sound synthesis research began there in the mid 1950s] working with Max Mathews—

LARSON: Right.

VERCOE: —essentially on the MUSIC V text. Yes, he was an innovator in that sense and really cataloged sound and continued to be a, sort of an expert in cataloging sound, and timbres, and things like that, along with [David] Dave Wessel [Professor of Music at University of California, Berkeley]. When I had worked with him a little bit at IRCAM—but he was then at the point of moving out and going down to Marseilles, as I said earlier—I was quite enamored of Jean-Claude and his—his ability.

He's both an excellent musician and an excellent scientist, and you rarely find that as a combination. Usually, people are good at one and not the other. But to find someone who is good at both and continues to practice both, and not simply lose sight of one of them, which is sort of what I've done, I perhaps think, Jean-Claude has just kept up composing and kept up scientific research. I sort of admired that whole thing, which is part of why I then had him come here to do a piece.

Now, just prior to that, when I'd been at IRCAM I had taken a student of mine, Miller Puckette [MIT S. M. 1980], who was a PhD student at Harvard at the time—he had been a math major here at MIT—I took Miller with me to IRCAM the second time I was there. And—so he lived at my house. And, you know, he was a—a big help in what we were doing. Except what he did there was in looking at how I was going about this "music minus one" thing, the idea of tracking the flute and doing—in the second year, it was actually tracking a violin, since Larry Beauregard had already died. So I was at this point tracking a violin, and doing automatic accompaniment, and looking at how the computer could be made to respond to *tempo rubato* performances, and so looking at, now what happens when two performers are coordinating in that manner.

Miller Puckette was in the background looking at how I was putting these together and started to come up with graphical representations of what I was doing.

And that ultimately led to something that was called Max/MSP. It came out of Miller looking at my stuff, coming back here with us, and doing something.

So beginning to—to realize that in a more formal system, and then ultimately going back to IRCAM and completing that Max/MSP work. And it was that Max/MSP system that then Jean-Claude Risset used in his *12 Piece—12 Etudes for—for One Pianist*. [Ed Note: *Eight Sketches: Duet of One Piano*. 1989.] And that was a wonderful example of getting into real time.

A computer was involved again and this time not in making the sound, but in simply controlling the relation between the live performer and the synthesized, that is to say, piano-controlled second part. And Jean-Claude worked out just a number of wonderful ways of controlling what the second piano did. Sometimes he would play a note softly on the keyboard, and this would do something fast on the—or he would play something loud here, and that would do something slow on the second machine. So he had various alternative mappings from one instrument to the other.

The greatest performance of that was perhaps when we induced—well, the first performances were done on two Yamaha uprights, the recording pianos. But later on, Yamaha came in with two grand pianos, big nine-foot grands, and put them into the—the hall, the performance place in—at—at—at the Media Lab. And so we had—you could clearly then see what the one piano was doing, and the other was—what Jean-Claude was doing and the other person—the other keyboard was doing in response.

We did a re-performance of that work later on, in a concert about 2000—yeah, about 2001, or something like that. It was a concert—

LARSON: Right. That was that celebration, 25th anniversary [Digital Rewind: a concert and symposium in celebration of the 25th anniversary of the Experimental Music Studio; May 21, 1999].

VERCOE: Yes, 25th anniversary. Yeah, we resuscitated that piece and, again, got the same sort of feel as just—and in recording that, we could then, of course, put one—one of the pianos on one channel and another piano on another channel. So if you hear that piece through stereo headphones, you really get a sense of who's doing what. And that was sort of a new, wonderful way of experiencing the two-piano, sort of, stereo effect of one live pianist and one following pianist.

5. Graduate students and technical innovations (43:07)

LARSON: Mm-hm. So with the Media Lab you have worked with a number of graduate students doing lots of different things, not necessarily music related, but obviously sound related, but some interesting things about, you know, sound perception and things like that. Was that, kind of, a natural thing for you to, kind of, move into some of those—those ancillary, kind of, fields with the graduate students?

VERCOE: Oh, natural to this extent: it was a natural evolution. I was interested in sound reverberation, and so forth. And I took on a former MIT graduate, who was at the time working for Kurzweil [Technologies, Inc.]. His name is [William] Bill Gardner [MIT M.S. 1992, PhD 1997]. I took on Bill as a graduate student. He wanted to do some work in—in—I suppose in sound development, particularly reverberation and—and things like that.

And what he then got into was something that you could describe as two-speaker 3D, where you have two loudspeakers that are, perhaps, at a workstation where you're sitting in front of your screen. You've got two speakers left and right, who are then sending sound to you. And if you do this correctly, you can actually position this—the sound object and create objects to the left or to the right by just simply having the sound from one speaker that reaches one ear, then canceled by the other loudspeaker before it reaches the other ear, but replaced by something that reaches the other ear a little bit later.

And that gives you the impression that the sound is off to the right, way off to the right. So, I mean—or—and so you begin to play sound around the room, 360 around the room, and even overhead, if you take into account the pinae effects. And—so Bill did some wonderful work on that.

I had initially heard this idea realized at a studio in the Roland Corporation research lab in—in—in Japan. And they were producing CDs that could have sound surround the thing. But in order to hear that, you would have to sit on the perpendicular bisector between these two loudspeakers.

And you sit there, and if several people wanted to hear the same effect and hear the sound throughout the room, they would all sit in a straight line, chair one behind the other. Because they all had to be on the perpendicular bisector between the two speakers. And that was wonderful, where I had a demonstration CD. But it wasn't a very practical thing to do.

Likewise with Bill Gardner's effect of having these two-speaker 3D things work and play some sound around the room, you either had to sit in—one behind each other, or if you had separate earphones—No, you wouldn't have earphones. You didn't want earphones. You would just want—coming through the speakers.

You could—it could be you're experiencing the same thing. But if you then—by sitting in—in this—in this particular place, but sitting stock still and not turning your head. Because as soon as you turn your head, you've then destroyed the sound, the internal time differences. And then the image was shattered. So the problem there was you had to sit still, you couldn't move around in your seat, and it wasn't a very conducive way to experience music.

But then Bill had the idea of having the computer understand what you were doing. So he put a head tracker—it was a Polhemus head tracker on the—on the user's head—so the computer could then tell if you have moved your head around this way or this way. And it would immediately make up—realize that your head was—your ears were now in a different location. And so it could do this same cross-talk cancellation, which is essentially what it was. And the image would stay. And that was a wonderful piece of work.

And so I was interested in carrying on that work. Bill did that as his PhD dissertation. He then graduated in 1980—'94, I think it would be. It'd be '94? Yeah.

And I needed another person to come in and do something to pick up that work, do something similar, carry it on a little further.

And that's when I chose a student who was actually a Boston student, but he had gone and done some work as an undergrad at Northwestern University. He had quite a lot of experience in au—in acoustics. And his name is [F. Joseph] Joe Pompei [MIT PhD 2002]. And he came in as a grad student with me. And he'd been—so he seemed to be well equipped to pick up from where Bill had left off. And I was quite confident that we were going to get another—the next step in this thing.

But he'd only been with me for about two or three weeks, may—maybe a month or so, when he said, "Well, now I'm here, let me tell you what I really want to do." [laughs] "What I want to do is create a loudspeaker that just plays like a laser beam, plays sound in one direction. So you could hear it, but you would—you could hear it, and they wouldn't hear it over there." So it just—

And it sounded totally unusual. Because loudspeakers, as you know, are like incandescent loud bulbs—light bulbs. They just send their energy in all directions, pretty much evenly.

But the idea of a loud—of a loudspeaker that worked like a laser beam was something very novel. And this required the sound to be encoded on something that had a much narrower focus, in this case was ultrasound. And if you—the degree of spread of ultrasound depends on the ratio between the wavelength of that signal and the size of the speaker. And so by controlling that, by having the speaker rather larger, as a sort of flat surface like a pizza box, pizza dish, and having the—the—the ultrasound up at around 60 kilohertz or something, then you could actually have a fairly narrow beam that was about—had a spread of two or three degrees.

And the audio, then, was encoded on that, sort of like an AM/FM encoding on the laser beam as the carrier signal. And then—you would then have to somehow decode that. Well, it turned out that 60-kilohertz ultrasound finds the air a very unstable medium, and it begins to break up, as soon as it gets about three or four feet away from this loudspeaker.

The loudspeaker was actually a collection of 70 or 80 little speakers about this size that were initially taken out of the Polaroid Land Cameras. This is the range-finding parts of the Polaroid Land—And so they had a high frequency response, good, but almost no lows.

Anyway, the 60-kilohertz carrier wave was fine. It was high enough to be above the threshold of hear—hearing, so that it—of human hearing—so it—it wouldn't do any harm to be hearing this. It was even above the threshold of hearing of cats and dogs. I'm not sure—quite sure about crickets and other things. And we'd never, sort of, got to finding that out.

But this idea of encoding the audio on a—on a 60-kilohertz ultrasound carrier was really very novel. And since the carrier then began—would begin to break up about three feet—three or four feet out from the speaker, what would happen is that the energy that was encoded as audible sound in that carrier wave would then began to fall out. And as you're app—experiencing this carrier wave passing by you, what you'd be hearing is the effects of this encoded audio falling out as a byproduct of the carrier wave breaking up. And you would then be hearing this sound just right around your—your head.

And it was just—it wasn't stereo, it was just a single-dimension sound. But it just gave the experience of having—being able to hear something for a moment as the—as the beam passed you and then not hearing anything after it had gone by. And that was just a—a wonderful experiment.

Now that is something that I would never have thought of as a faculty member. So for those research groups where the faculty had most of the ideas, and the students were there just to simply realize these things, they probably wouldn't have come up with something like that either. So I had great faith in the ability of the younger students to really innovate and be creative.

And Joe [Pompei] was one of those. And he had the wonderful idea. He's also very—technically very solid. And so he came up with this idea, realized it himself. It was a very expensive piece of development—research and development.

But it was one of the best demonstrations we could then give at the Media Lab. And it was one of our standard ploys for quite a few years. And you've probably heard that.

LARSON: Yeah.

ARIZA: In terms of, sort of, more in general, as an advisor, as a mentor for these students who are maybe doing things unrelated to your main area, how did you, sort of, nurture the kind of creativity that seemed to define, in a sense.

VERCOE: Well, yeah, I see. I had one rule. And that was I'll do my work, and you do your own. [laughs] And that meant the way the student group operated was like this. I would—I learned this from Marvin Minsky, actually. He'd say, "Barry, the way to proceed here is you just get a bunch of bright students, and you just put them in a room, and close the door. And then you just try to give them what they need, and then try to stay out of the way."

And that's—was Mar—always Marvin's idea of how to direct a student group or to be an ad—student adviser. It let them be the creative people, in other words. And I had that attitude, rather, that I would tend to just do my own research. And not—and nobody, none of the students, did my work. And I didn't do theirs. But I would sort of enable them to do theirs.

Now that be—that depends on having the right students. And choosing the right students was something that I did the following way. Whenever I would have applications—there would typically be 30 or 40 applicants to come into the—into this music group at the Media Lab each year. And I would go through all the applicants and pick out about 10 that looked likely.

And then what I would do is hand out copies of not the entire CV, but typically the personal statement that these applicants had written. I would hand out copies to my students who were staying on and say, "What do you think of these?" And what they would be looking for is people who would be good office mates, people they could work with or something. And so they would be a large part of the decision making. And in the end, we'd invite perhaps two or three of these people to come by for interviews. And they would take them out to lunch, and so forth.

But it was those students who basically chose their colleagues from year to year. And that has given rise to an alum collection from the Expiri—from the studio there, in which everybody just likes everybody else's work, and they remain friends. They meet every, you know, a couple times a year for a beer or something like that. They remain very good friends. And there was never any real, sort of, conflict or—or, should I say, competition between them. They were all doing work that they respected of each other. And they have all remained very good friends.

And I've found that, for the most part, they were right. I might have chosen other people, when it came—there was only one instance where I overrode [laughs] them and I said, "No, I think you're wrong. This is the person that—" And that was when a student by the name of Paris Smaragdis [MIT PhD 2001], who had come out of UC Berk—not UC Berkeley, Berklee College School of Music across the river here. And he didn't seem to impress everybody or anybody. And they thought, "Oh, he'll never make a researcher."

I—I overrode that. And I said, "I think you're wrong on that. I think Paris has the makings of a researcher." It turns out that Paris has been one of the best researchers we've had in the—in the—in the Media Lab, for the work that he did on blind source separation and things like that, where he really had an insight into how sound worked. As a musician, he was—had a great ear for that sort of thing. Didn't have a large background in—in mathematics, but was very adept and quickly picked the stuff up, and turned out to—to be a wonderful member of the group.

But he was the only person that I chose. Everybody else was sort of self-chosen by the group. And I think it—it did work out very well.

7. Csound synthesis software, origin and developments (56:37)

LARSON: So, I want to get into—give you a chance to talk about Csound. And that's one of your, you know, lasting legacies. And we're coming up on a break here soon. But do you want to talk about, kind of, the—the origins of Csound and, you know, and how it's related to the earlier stuff you were doing?

VERCOE: Yeah. Well, it came about in a couple ways. First of all, in the early days of the studio at MIT here, the Experimental Music Studio, we decided to install UNIX as our operating system.

Now everybody else in the—at MIT was running a thing called either RSX or RT-11, which both were operating systems from Digital Equipment Corporation. If they had Digital machines, they were running RSX and RT-11. And I decided to switch to UNIX in the very early days of the studio because in—I was interested in the—the structure of C, which was the language in which—used UNIX.

And so it was C and—and UNIX were—sort of came together as a package. And that enabled me, then, to write Csound a bit later. But the first thing was to establish UNIX as the operating environment here.

Now, that initially put us out of step with the rest of MIT because the rest of MIT was able to send messages to one another, you know, e—primitive email, sort of, communications and so forth, using RT-11 and RSX communications. And it just wasn't connecting to UNIX at all. So to have the right idea at the wrong time is not always a good idea. [laughs] So, we were sort of off by ourselves.

And I had signed—I signed the initial agreement between MIT and Western Electric, which was—had been handling the—the patent things of Bell Labs, Bell Telephone Labs, for the UNIX environment. And so for a year or two, I was the only signature on this thing. But later on, when the rest of MIT began to see that UNIX was the—the thing to go with, then they just simply appended their own signatures, so—but I was, sort of, the—the initial UNIX thing here.

Now that had me working fairly closely in—in—in C to begin with. But I was still running MUSIC 11. Because on the PDP-11, even though we were running Linux, I was—MUSIC 11 was running in assembler language. So it was RT-11—no, sorry—it was PDP-11 assembler. And so I was cranking a lot more speed out of the—the machine, a lot more com—compute power out of the machine, by sticking to the assembler language.

I had done the same with MUSIC 360—that was all assembler, too—and could beat the F—the Fortran equivalents by a factor about four or five, or something like that. So in the days when computer time was very expensive, that was a big advantage, for s—for a composer to get five times the amount of music out of running the MUSIC 360 assembler version of what later became MUSIC V or MUSIC IV-B, which was the—the—no, IV-BF, which was the Fortran version of—equivalent to MUSIC IV.

So coming from the PDP-11 assembler version—sort of a version of Csound, with the one difference that I established in—in PDP-11—in MUSIC 11, the idea of control signals. In MUSIC 360 everything was running at the same audio rate, controls and everything else. In PDP-11—writing MUSIC 11, I decided to establish another network, I suppose, internal network of signals, which were the control signals. And they were—I gave them little names of k-something. So audio signals were asig, and control signals were ksig, where—where sig would—might be some other name. That I actually got from [Donald] Don Buchla.

Buchla's synthesizer was different from the [Robert] Moog synthesizer, in that Don Buchla had two classes of patch things—they were both analog things—but two classes of patch cords. One of them were audio cords and the other color was control cords, control signals. And so these control signals actually controlled the analog oscillators that were putting out, you know, full-frequency audio signals. And so he had control structures and audio structures. And I thought that was a great idea.

Don Buchla's synthesizer never made it to prime time, but was a very interesting thing. There was a lot of these things around at the time. A lot of music came out of that, but never became as popular, perhaps, as the Moog synthesizer. But

I thought the simplicity of having control signals and audio signals was a really amazing step forward.

[break]

LARSON: So you were talking about Don Buchla's patch cords—

LARSON: Hmm. Yeah. And the—the beauty of Don's system was that it really did recognize, whether Don knew this as a—from a science standpoint or not, recognized the difference between our perception of frequencies and our perception of envelopes. Envelopes operate at a different rate. We can only hear, like, 20 consecutive sounds, or 15 or 16 sounds, as independent entities before they then become—start to become pitches. And that's—that depends on the rate at which neurons can—or the envelope, sort of, detectors can actually res—be reset and accept another sound.

And that's a very different thing from perceiving pitch, which is where, sort of, now looking at how things operate at a different—at a different speed. Now the fact is with pitches, we don't really hear the pitches because the neural rates and the auditory mechanism don't exceed more than about 1,000 times a second. So we're not actually hearing anything over one kilohertz. What's happening is it's energizing the part of the sensory mechanism that we associate with those higher frequencies. But the—the energizing is not active—actually making those things fire at that rate.

And so there's a difference between the way the—the auditory mechanism detects envelope shapes and the way it detects pitch structures. And Don, whether he knew that or not—as an engineer he possibly did, or sensed it in some capacity—had neatly divided these two things. And I thought that was a nice division. So when I came to write MUSIC 11, I actually put in this other level of control, which was the—the control signals that—that would control audio oscillators in the language.

Now throughout that, in writing MUSIC 11, I had—had various experiences, when I was writing "Synapse," of the shape of envelopes and so forth, that initially our idea of an envelope was sort of like a piano envelope that would, sort of, rise and naturally decay. Whereas—I think I may have told you before—my violist could actually have much more control over the steady state of the note. And so I then came up with different-shaped envelopes that would rather emulate what string players and wind players could do that pianists ordinarily wouldn't do. So that gave rise to a different kind of control signal requirement anyway.

But in writing all of this stuff, I—I would write these things in assembler language, as I said, for speed. And I would carefully document with little comments to the side, you know, "Add this to this," or whatever it is. And I'd—so I—I had this—my own notation for documenting what was—what was—wasn't so apparent in assembler language. It was a little bit obs—obtuse.

But when it came to writing—but—but I should say that the documentation was fairly complete. There'd be an oscillator. Everything would go—the filter, everything, was—was encoded. And, you know, four times the square root of this or that, or whatever, was all right there in the comments.

So what—how did I write Csound? I basically took MUSIC11, deleted the assembler language and compiled the comments.

LARSON: [laughs]

VERCOE: And that was about it. There had to be other changes, of course. But that's, sort of, how I got into the C version. Because it turned out that my comments were largely C-oriented, C-based. So C is a natural, sort of, set of symbols and operations and things. And that's worked pretty well for my—so I had to do a little bit of tidying up, but that sort of how I got there.

That's not giving C in real time, not real-time Csound. That was a later thing that required computers to get a bit faster before that could happen. But that's how the earlier version of Csound actually came into existence.

ARIZA: So were you able to get the performance with that move to C that you had with the assembler code? Or close enough?

LARSON: Well, I was on a different machine now. Because I was actually moving on to—well, I was still on a PDP-11, I should say, floating-point processor, yes. But I was moving towards a VAX machine, which is where—that had a lot more speed than PDP-11.

At this point we'd had—It was given to me by Digital Equipment Corporation in 1972. And we're talking now about this thing happening in 1986, '87, when—I think it was around that period that we took—a V—took delivery of a VAX machine. So by the time we moved to the Media Lab, which was 1985, yeah, we already had the VAX, I guess, then.

So the VAX had come along. So we actually moved the VAX into the Media Lab along with the PDP-11. The PDP-11 stayed on board, stayed online, as a generating machine for another decade. It was an amazing piece of—piece of equipment, best machine that Digital Equipment ever made, the PDP-11.

That was all due to Gordon Bell, who was the innovator there. The PDP-11 had been his, sort of, invention the—the Unibus that was—was based on—had been Gordon Bell's innovation. And he, well, then became the director of engineering at DEC. But that work had been done, actually, I think at MIT here.

ARIZA: And then the move to real-time Csound happened—you talked about using it in—by 1990.

VERCOE: Yes. Mm-hm. 1990, that was the first demonstration of that. So all I had to do there was to replace the circular flow of control. And in both MUSIC 11 and in Csound, there were control signals.

There was one pass through the code at the control rate, looping through the code at the control rate, and that each time it got to an audio-generating signal, it would put out an array. So it was array processing the audio signals and then some single—single-step—well, single integer or—or part of the audio signal. So there would be one audio word would come out, and then an array of words of—of the audio signals.

And that loop, then, would just, you know, would go as fast as it could. lf you had a big, heavy piece of music, then it would slow down and would gradually—but it was just writing onto a—a digital disk anyway. And it wasn't att—attempting to be in any particular rate, except just to be true to the score.

What I did in real-time Csound was then put a monitor in here, in this loop, that would constrain how quickly it got back to the—to returning to the top of the loop. And that was like having a—a governor on a car—if—I don't know if you know what a governor on a car is, it sort of prevents it from going over 50 miles an hour or something. This is what the—the car rental companies used to do to— [laughs] So I had a, sort of, a governor here that would be just waiting on something before it would go back and—

Now this "something," waiting on something, the something was the ability of the D-to-A—DOA converter to actually empty out a block of buffered sound. And so this was essentially writing to the—to the DOA converter, sort of, a—and blocking on the output. So it would be waiting on that. And the—as soon as we delivered another block of sound, then it says, okay, back to the top, and do it—generate some more. [Ed. Note: D-to-A, DOA, D/A, and DAC are all abbreviations for "digital to analog converter."]

And there'd be typically three or four buffers that would give you a little bit of take-up in case you got behind on the—on the sound generation. Because sometimes the big, heavy things might have been slower than real time. But in general, you—by using this buffer—a buffer of two or three buffer blocks, you could actually keep going with fairly complicated signals operating in real time.

So that's basically what I did for real-time Csound was just insert these little sound buffers in there waiting on the output. So it was—it was buffered or—on—it was blocked on I/O [input/output]. And so you're just writing in that. And that gave real-time Csound. And that's what—so I didn't have to do much there, either.

LARSON: So with—aside from the real-time thing, were there particular musical problems that were, kind of, underpinning, kind of, what—how you structured Csound, and the fact that you had, you know, these separate audio and control signals, that gives you a certain kind of control over the kind of sounds you're, you're trying to create? But, you know, the—talk about some of the underlying kind of musical premises for the—the engineering work.

VERCOE: Hmm. The idea was to, first of all, produce very efficient oscillators and filters that were capable of running in real time and not simply taking so long to deliver sound to the output that the buffer would empty out. So each orchestra comprising a number of instruments, or so, that may be all operating at the same time or something, or perhaps just one instrument at a time, or whatever. Each orchestra would have this finite limit of how much it could get done before the—the buffers would empty out.

And you'd—and you get a stuttering effect, and—when the deck would repeat a buffer, before it was refreshed with a new block of sound. And that was sort of dut-dut-dut. You—it—actually, you could tell when the—it was getting behind real time.

So that meant that we just simply had to make it very efficient—a very efficient orchestra. And so it stayed in assembler. Or in the case of C—the beauty of

C language, as opposed to other high-level, C++ and other level—other languages, that C was running very close to the hardware.

And I was sufficiently familiar with what the compiler was going to do to my C code to know what assembler it would generate. And I would use this op code as it—or this—this C structure instead of another C structure, because I knew that this one would actually run faster. And so with—by keeping an eye open for things like that, you could still cram a lot of speed out of this thing.

Now that's just dealing with technical things. But the idea of getting into something that was now real time meant, as I was saying earlier, suddenly you had an interaction with the computer at the speed of the piece that was being played. And that meant, first of all, that the computer could listen to your input and exert another element of control over this looping stuff. So it wasn't just simply the sound being buffered to—to—through—through these buffer blocks. And that, sort of, was controlling the rate at which it went through the piece. But the rate at which the events of the piece occurred, in other words the tempo of the piece, could be then controlled by something else.

If the tempo was slow, of course when you slowed the tempo down, then there would be more buffers of output on this particular one note because we are now operating—playing it slower. And then if you speed it up, there'd be fewer buffers, of course. So the—the d—the digital-to-audio conversion [DAC] was still the only clock in the piece, in the whole system. So the DAC was the clock that controlled everything.

But over and above this digital audio rate, you could then impose any kind of tempo that you wanted. And the tempos were running asynchronously with the digital audio conversion rates. So that was the fixed rate. And then over and above this, so once you'd—you had a fixed rate that was giving continuous sound, you could interact with that. Then you could do things like tap a tempo with something and have the music responding to some tempo that you wanted it to do, or have the computer listen to somebody playing something or tapping on the desk.

In fact, the first demonstration of real-time Csound, which was done at this ICMC in Glasgow, had one of my students, Dan Ellis [MIT 1996], who's now teaching at Columbia—he was a—Dan was an amateur percussionist. And he was just tapping on the desk here. And the computer was figuring out from his tapping what the tempo—the tempo was.

Now that was an interesting musical challenge, to have a computer have a sense, from almost random tapping—or unstructured, should we say, largely unstructured tapping on the desk. You know, just [tapping]. Nothing that was previously scored, somebody just tapping. And, you know, when you and I hear somebody doing that, we infer from that a certain tempo rate.

Now how do we do that? Well, the tapping is coming along in some sort of little patterns of *dump, da-da dump, da-da dum*, or whatever it might be. And you infer from that there's a, sort of, a metric. So the idea was to have the computer able to infer a metric from some semi-structured input—as you and I might when we're listening to some—somebody just tapping away—and then for the computer to use that as the tempo required of the piece that it's actually generating. And that gave Dan Ellis, then, the ability to speed it up and then slow it down just from—by increasing his inferred tempo rate in his random input.

Now that was new because we'd never had anything like that. Previously if you wanted to speed up, you'd, sort of, use a potentiometer and, you know, just jack it up or pull it down. But to be able to just do it, control it, from essentially musical input—musical event input—was a new thing.

And you could only get into these things when you're in real time. There was—you couldn't do anything, realistically, with that kind of input before it—real time. Or you could, I suppose, and it would be just come an—it would become an academic exercise.

But once you cross over the real-time threshold, then you're into the real—real-time interaction. And that turned computer music into a very, very different thing. So the demo that we put on—we gave at the ICMC in Glasgow in 199—I believe it was 1990, just put computers into a totally different domain. And that was a big breakthrough. [Ed. Note: see page 3 of this transcript.]

ARIZA: Mm.

LARSON: So Chris, you had some other questions about Max/MSP and Csound, and—

ARIZA: Yeah, sure. I mean, the—the interface of Csound and the interface of computer music languages I think is a really interesting question. You mentioned Miller's work in the development of Max/MSP.

VERCOE: Mm-hmm. Yes, yes.

ARIZA: And, of course, there's PD [Pure Data Computer Music System], as well. These visual programming languages offer this visual analog to the old patch bay and provide a certain level of accessibility, whereas text-based languages offer us a certain amount of power. I wonder what you feel about, you know, these visual programming languages and what they offer computer musicians.

VERCOE: Oh, I—I feel they're—they're a great addition. I've never got much into it myself. I've just left that to others. And that's been just fine. So the various people that have come up with front—front ends for all sorts of things—front ends for MUSIC 11, front ends for Csound, et cetera, and I've been happy to see those things happen. Because they have given the users a much more intuitive sense of what—what's going on. And I don't have any negative feelings about that at all. It sort of increases the user base and the ability of people to actually work with the material.

I have stayed rather true to the integrity of the code underneath that does the sound generation. Because I've always regarded the quality of that sound as being uppermost. And you can only get that by focusing onto those things and not getting too distracted with whatever the—the pretty things are on top. [laughs] I prefer to get my kicks from the beauty of the sound that comes out, rather than the—the fancy, sort of, front ends. But I do agree that those things have been really a great addition. And that's been great.

ARIZA: At a certain point, the—the licensing of Csound changed, maybe it was 5 or 10—10 years ago or so, in that range. And it became an LPGLed [GNU Lesser General Public License]—

VERCOE: Yes.

ARIZA: —license system. Can you talk a little bit about that licensing change and how that, sort of, promoted the growth or distribution of Csound?

VERCOE: Yeah. Well, all along, I had had the habit of just making all my code public. And MUSIC 360, in the beginning I would be mailing off, running out of the post office every second day and mailing out a—mailing off a 360-foot—360-foot reel—or 300-foot reel of tape to people who wanted to be running MUSIC 360. And the same is true of PDP-11—the RT-11—not RT-11, the MUSIC 11.

I would happily give people access to these. In that case I was able to send it over the—the Internet. So that was a lot easier than running down to the post office with a 300-foot reel of tape to mail off to somebody.

So I've always given things away. So I was quite happy for people to be adding, when they got hold of this, to be adding their own things. Jim Dashow, for instance, even in MUSIC 360, [James] Jim Dashow wrote his own op codes.

He's the only person, I think, who's ever done that. But he was sufficiently into the code and able to, sort of, envisage some new thing that he wanted that I hadn't supplied. And he would sit down and figured out how to write the new op codes, which was amazing—in assembler language, you know—360 assembler language, that was—that was written in. And, yeah, Jim's the only person who, to my knowledge, ever did anything like that. But I was happy to see him do it and add his—and just enrich the environment to—to his own needs.

And people have continued to do that. With MUSIC 11, they would, sort of, write out their things—not only front ends, but actually get down and write, sort of, new operations down at the guts of—of things. And that continued to happen with Csound. People wrote front ends. And also people would add op codes.

It got to the point where the composers who were writing op codes were doing it in resp—in response to what they felt their own needs to be. I should say that in the case of the expansions of MUSIC 11, new op codes there that came rather out of my summer workshops, when people would say, "Oh, I'd really like something that did this. It doesn't do that." "Okay, I'll prov—I'll do it for you, and you'll have it tomorrow." Or in my own case, I needed something that did this a little differently.

So it was usually these op codes would come out of the creative need to solve a problem that you'd, sort of, confronted yourself. [laughs] It was your own fault, but you had to solve it somehow. So you would typically come out with another op code that's—that would solve—do this—that thing.

Sometimes it might be just a case of efficiency. You could with the existing op codes do this whole—create this structure that would solve the problem. But then it was too slow to have all these things up. So you would write one op code that did it all efficiently, where all the—sort of the—the minor details were inside the single op code and could be all done in assembler, as opposed to passing, like a bucket brigade, between op codes.

And people were doing this through the '90s, in fine style, and then were generating pieces, and putting pieces on concerts, and then wanting these pieces to be recorded. And you know, that was just fine by me, since I had no qualms about them recording their own music, *et cetera*. But there began to be concern, because Csound had NSF written all over it. That is to say, I had some support from the National Science Foundation to be—to be writing that early stuff. Even though, as I said, I just simply compiled the comments, [laughs] there was some support from the National Science Foundation.

So it—it had to be with, you know, a reference to National Science Foundation. And maybe there was some—there was concern, there were some reservations about that, to what degree were these things that people were now putting on top of my Csound. Were they owned by the composers? Was their piece now owned by the National Science Foundation? Or whatever. And it was a legiti—legitimate concern, I felt. And so by moving to an LGPL [Lesser General Public License]—it's sort of a more open-source or more available agreement, then people felt that they could actually write and produce pieces, and write their own additions to this thing, and proceed and—without concerns about suddenly violating some rule somewhere.

Once you do that, of course, you then open up the thing to—to all and everybody to start adding things. And sort of the community of Csound is around the world. I've—I was—I'm always amazed at how large that has become, through the—the push that has come from [Richard] Rick Boulanger over at Berklee [College of Music] and lots of other people around the world who've—John Fitch in England. There's a huge community out there. So there's a—there's a Csound community. And, naturally, everybody wants to be in there and adding their own little op code, or something or other. And there was nobody really riding herd over that, unlike in the case of—of Linux, where Linus [Torvalds] would say—he'd—he'd, sort of, act as good cop, bad cop or something and say, "Yes, we'll accept that, we won't accept this." And that sort of kept a lid on somewhat. Csound got a little bit out of control.

I wasn't controlling it, and the people who started out controlling it, notably John Fitch at Bath—University of Bath in England—gave up control. He was a math professor, and he had other things to worry about. And so that the public Csound became quite—increasingly larger. And that eventually leads to software bloat, that systems become just too big to be maintained or to be useful on a small machine, or something, fine on a great big Intel machine, these days, expensive machines. But if you wanted to get into smaller machines that you wanted to use in concerts, and having several of these communicating with one another, a big system like that becomes a bit of a liability.

And so I was happy to see it happen, under LGPL, that people could do all this stuff. But I always maintained my own version of—my, sort of, private version of Csound. So there's a public version, public Csound, and my own version of Csound, which is what I was using for teaching over here. It was something that I could more easily teach the classes with. There was limited variety of things that people could do when they were just learning about this stuff. And I've been happy just to run my own ship, as it were.

And it's just like doing my own research, as I said earlier, doing my own research and letting people—other people do their own. [laughs] And never the twain shall meet, or didn't need to. And so I've just let the public Csound just go along on its own course. And I've, sort of, maintained my own thing.

Now the disadvantage, then, is that I haven't had the use of some of these other interesting innovations that have come by. And as I haven't tried to emulate any of those things—a couple, but generally I just kept, sort of—kept to my own more constrained environment and selection of op codes. And so I've—I slighted myself, I suppose, in that sense that I just hadn't wanted to get involved in these other things.

But it has meant that my own little version runs easily on some very small machines, like the XO, for the One Laptop per Child thing. And so I haven't, sort of, been too concerned. There's certainly a lot of flexibility and enough power there for the kinds of simpler examples of computer synthesis that would come up on mach—small machines.

ARIZA: Yeah. The—one of the amazing things about Csound is that it runs on nearly every platform, and continues to run on every platform, and has been used and embedded in—in many, many different things. The one you mentioned there—the One Laptop per Child program, when I first saw that there was Csound in that, I thought that was—that was just amazing. It also has a novel, patchable front end, as well, which I was interested to see. I was wondering if you could talk a little bit more about the idea of giving synthesis control in a—a system like that and the ability to synthesize music to a device that was initially intended, if I'm not incorrect, to children in Africa, really, and around the world.

7. Education in remote and Third World countries (88:40)

VERCOE: Yeah. Yeah. Well, you then run into another set of goals. The One Laptop per Child was indeed created by Nicholas Negroponte [Chairman Emeritus of MIT Media Lab] as a—as a—with the objective of providing inexpensive computing for disadvantaged children around the world. Initially this was Rwanda, in the middle of Africa. And later Rwanda sort of fixed itself.

But there are certainly other, many oth—many other communities of kids around the world that are without advantages, a lot of them being in South America and Peru, and so forth, where the government has seen fit to purchase large numbers of these XO machines for the kids in the re—very remote areas. I took some of these XOs initially down to the South Pacific, which is where I'm from. Initially it was to the Solomon Islands. I had a friend there, someone I'd met online, basically. And he was asking for some XOs.

And I managed to get hold of thirty machines. And I sent thirty machines down to this person, David Leeming. He was a—an English—Englishman who has been teaching in the Solomon Islands, in fact married a Solomon Islander. And they've got a family there. And David's an excellent teacher, but also was getting very interested in—in the technology. He's not a composer, but getting very interested in the technology that could enable the children of the Solomon Islands to get into—in to have a different environment for learning.

And so I sent him thirty machines, thirty-five machines, I think it was. And after six months, I got on my bicycle and went down there and looked—went to see what was hap—happening. I had asked David to put these machines in the most remote place he could find, where the children, by implication, would be the most disadvantaged, cut off from the rest of the world. And he did that. He put it—he had put them in an island of Patukae, which was way off to the western part of the Solomon Island group.

And when I got there, landed on Honiara. I then took a three-hour boat ride on a, sort of, river launch kind of thing around some of the islands and got to this remote island. And then from that one wharf, then we took a dugout canoe for another hour and eventually got to the school, where they had some of these XO machines, the first XOs being deployed in the South Pacific.

And I went to see the—the school. David was with me. And encountered the school here, where Elnah [Tati], who was the teacher—very good teacher, actually, and was thoroughly in command of what was going on. She had all the students all, sort of, sitting there along a—on either side of a desk, a long desk. And they were just working quietly away. She was—window dressing. I mean, she knew I was coming and so she had all these students, sort of, sitting—They were all about, oh, 14 years old, or something like that, and both, you know, boys and girls.

And I said, "Elnah, that's fine. But let's take half of those kids out and let in a lot of little kids, who haven't so far had this exp—And then let's stand back and see what happens." [laughs] Because this machine and the software had been designed, really, for younger children, for, you know, 5 to 12, or something like that. And so we did this and then went on a little walk around the playground of the school, the field and came back about 10, 15 minutes later. And the decibel level in that room had gone way up—kids teaching kids. And there was just an excitement about it that was really a pleasure to see.

And later on, we then took these machines—the—had the kids take them outside. And they were, sort of, you know, sending messages to one another, since these machines communicate with one another. They were having a lot of fun with that. It just gave me renewed faith in the idea that kids can learn very fast, perhaps in—with the help of a teacher, though not always necessary. But this machine was certainly in—giving these kids a new experience. That seemed to work pretty well in this one very remote school in the western islands of the Solomons.

I then came back here and said to my colleagues, Nicholas and the others in the One Laptop per Child project—they were focused on Rwanda and these other places. And I said—I said, "But let's—let's take some machines down to the South Pacific." And I said—They said, "Well, there's no genocide down there." I said "No, but there are a lot of kids who are very disadvantaged. And remoteness in—in the

Pacific Islands is like remoteness in a desert. You know, you've got water instead of sand between very remote villages." And—

And so I was given 5,000 of these XO machines, to take down to the South Pacific, that came from the Give One, Get One program where they had sold about 90,000, I think, of these machines, and—in this Get One, Give—Give One, Get One program. And there—I was given 5,000 of these to take down and distribute around the Pacific Islands, which I did. The big problem with the Pacific Islands is the places like Papua New Guinea, which has got a population of—well, in the '70s when I left New Zealand, the population of New Zealand was 3 million, population of Papua New Guinea, 3 million.

Go back there now, population of New Zealand is 4 million, population of Papua New Guinea, 6 million.

ARIZA: Wow.

VERCOE: So the whole population just takes—has taken off because there's all these teenage, adolescent girls who have no future except having babies. And the population is just going to skyrocket. The only thing that will stop that is education. So, in various places around the world, it's—the two—the—the things that are missing are, first of all, contact with the rest—the isolation is a—is a big problem. And the other thing is providing these young, creative minds to get into something that has to do with education. So you've got to provide educational resources for these, to—to stop the growth of pop—

They either going to have big families, or they'll become terrorists, or something like that. It—it's the—it's the same thing. Education is the only thing that will stop those things, and that we have problem visited on the rest of us in this world before we know where we are. So education is the solution. And that's why at this point, having retired from MIT, I'm very devoted to the education of the people in the South Pacific.

I've taken it from Solomon Islands into the—first of all, I spent three months in Papua New Guinea, which is the big problems I was just describing, but also into the outback Australia, where there's a whole population—there's about, oh, something like a million aboriginals there, who essentially have been just left behind. You've basically got there a first-world country with a, I suppose you might say, a third-world or fourth-world appendage that's just not being catered to properly.

And so I've devoted a lot of my attention now to aiding and abetting this—the education of the remote kids. It's not ethnic. This has nothing to do with ethnicity. The poor performances on the NAPLAN. That's the, sort of, literature and—numeracy and literacy sort of testing. And these around the whole of Australia show—shows up that some of these communities do very poorly on the national—the NAPLAN scores.

But as I say, it's got—it's nothing to do with eth—with ethnicity. Because these Aboriginal kids are as bright as buttons. It's remoteness. That's the problem. And in a place like Australia, you can imagine remoteness can be pretty severe, [laughs] you know, where these kids are living in little communities that are a hundred miles from the next—from the nearest community. And so this is where technology can get in and enable them to become part of this one world before it gets too far away from them.

LARSON: So how are some of those kids using the Csound on those—those computers?

VERCOE: Well, the—there are some things that have come out from the public Csound community, of a group up at Toronto—or McGill, actually, have come up with a little set of things called TamTam, this, sort of, little interactive systems that the kids can use, and they get sounds back, and they can have some sort of control, do a little bit of composing. But it's more, sort of, signal processing and patching things together. They don't—so they're not really composing the score so much.

What we've done here at the Media Lab is to come up with another thing called the Music—a MusicPainter, where kids are given a canvas, and they can draw on this canvas, and draw faces and image—and patterns, and things like that and then hear the sound back. And that's C—Now that happens to be running my Csound, as opposed to public Csound. Well, it doesn't make very much difference. But when you give children capacities to do that, to be creative and come up with sound structures and—and paint things in sp—with—using spatial graphics, and then hear that sound back and get a sense of how space relates to—to time, then they're into a learning—interactive learning environment. And that's nice to see them doing that.

8. Reflections on career at MIT and future plans in retirement (100:09)

LARSON: So we're, kind of, running up against the clock here. There's some other, kind of, big topics, things about, kind of, music at MIT. There's a quote here from MIT President Howard Johnson back in 1971. He says, "At MIT we have disagreed with those who think that the culture of the arts and the culture of the sciences are separate and immiscible. We find a positive value in an educational program that seeks to give students an opportunity to understand and appraise, appreciate, and, in fact, perform in something substantial in the arts as well as the sciences." Can you just talk about, kind of, the musical then the other contexts?

VERCOE: Yeah. It was a nice, pleasant discovery for me, when I first came to MIT, to find that the—the arts were being seriously considered and—and taught, and so forth. Very different—arts and humanities, for instance, at Caltech [California Institute of Technology] were, sort of, very low on the tote—totem pole there. Here they were given a lot of support. There was an academic credit—in fact, the students in general were required to take one arts course each semester—arts being arts and humanities, history or something. And music became, sort of, one of the options. It was very nice to see that.

Moreover, the typical student here, typical one perhaps coming from the Far East—from—from the—from the East, from Japan or someplace—oftentimes those kids were very good violinists, coming out of that culture. They just happen to be also extremely good at nuclear physics. And that was clearly going to be their—their main vocation. But they maintained their interest in—in and practice of—of music. And that had led MIT to support the—the growth of the Music Department here and the growth of music programs, the chamber music program, for instance, where you've got 150 kids every semester are involved in—in chamber ensembles, string quartets, and so forth. You've got a symphony orchestra, et cetera. And MIT administration has seen fit to support that because they believe that's very much part of the rounded scientist, to have opportunity to be expressive and to be part of the cultural environment.

Now that hasn't always been the case at a lot of universities. And so I think to MIT's credit they have sust—sustained the music programs here very well. And that's often a surprise to other people, to—to feel that MIT, which is regarded as a technical place, as—has had such a—an active music program here. And that's always been a nice thing for me to observe.

And a lot of the kids who came through the early Experimental Music Studio had come from music courses and just were sort of doing something in writing for computer as just an extension of their own experience in the music programs here.

LARSON: So, if you look back at your career starting out as a composer and moving into engineering and software development, how do you, kind of, look back at that as—from your perspective now?

VERCOE: Oh, would I do the same thing again? Probably yes. I've often felt that I would just like to, as I've got heavily involved in something, I would just like to run away and write a string quartet. But I—while I certainly could have done that and done less of the technical stuff, I turned out not to have written many string quartets these days. As I started out saying, it is nice to have—it's rare, but nice to have people who, sort of, feel some affinity in—to—and competence in both areas, of both the arts and the—and the sciences. And that's sort of what we've tried to do here.

It's rare that you find someone who is really good at both. Someone, they usually find, is good at one and—and not so good at the other. But it takes both kinds. In the end, what you basically have in this whole, total environment, is you have system builders and system users. And that's sort of the way the world divides into two capacities, two territories.

And I suppose I've become a system builder rather than a user of my own systems. And I—but I have always encouraged other people to use the systems that I've developed. And I've encouraged those, you know, by running summer workshops, and finding support for composers, commissions for composers.

And I've just always believed that the world should have access to whatever the technology has—of this day and age has created, and rather than just being technology for the sake of technology or for the sake of other bad things. I think our—as—as artists, our obligation is to have the computers create music rather than do war games. And that's sort of been my philosophy all along, I think.

LARSON: Mm-hm. Do you have any future plans for writing some more music?

VERCOE: Not immediately, I don't really have. I mean, now that I'm living, for the most part, in Australia, I don't even have a piano in my house. [laughs] So, I haven't immediate plans to do that, no. But I'm now, for the moment, living in my house here in—in—out in Natick, where I do have a nice piano. And I—I'm surprised at how much I'm actually playing that these days.

LARSON: I know that you're not so much, you know, kind of, currently involved in the computer music community. But, what are some of the—what are some central, kind of, problems that you'd like to—to see, you know, tackled, as far as computer music and, you know, just furthering the course of music? What—what would you like to—what's—?

VERCOE: What I would like to see is more people who are musically adept and musically sensitive to things getting into assisting with the technical de—and technological development. I think there's a big opportunity for people from other countries, not necessarily the US, other countries that now have the—a fairly mature technical environment to actually get in and do the sorts of things that I've done. I'd like to see in, for instance, my own country of New Zealand, I'd like to see that happening in the universities. It hasn't much happened to date.

But I've just gone through a series of lectures to some of the major universities in New Zealand, encouraging them to do this. And I said, "Why don't you guys get in and—and write some—some new fascinating code or something, and have computers do some really amazing things that satisfies your creative and—and expressive needs?" And I think that's the sort of thing I'd like to see continuing.

LARSON: Mm-hm. Do you think it's possible that there's a kind of sound synthesis that's—that's not possible now that could create some sounds that we really have not heard before? Or do you think that the future technical advances will be dealing with kind of the sounds that we've heard? I just wondered if there's some new dimension that we—

VERCOE: I'm not fascinated with new sounds. I'm fascinated with processes that will take sounds that we are familiar with or can recognize and sort of modify them so that we can understand the modification. But just new sounds coming out of the blue don't particularly fascinate me. I'm rather more fascinated with the tran—the transition of sounds, the modification of existing sounds into another sound and say, "Wow, these are somehow related in some capacity."

But just the discovering of new sounds has no appeal to me, actually. I don't—I mean, because that's not musically motivated. I think modifying a sound to go from one kind of sound to another sound has sort of an evolution to it and can be part of a musical motivation. Just coming up with a new sound from out of the blue seems to have no cause or need to exist.

LARSON: Mm-hm. Mm-hm. And what's so interesting about your work is you're—it's always based upon a musical and a human premise. And that you're not looking for computers to replace—replace humans. And as—as we all know, computers are more and more a part of our everyday lives.

And some people even have this idea that music is only something that comes out of a loudspeaker. They don't even, you know, think about, kind of, human performance. But I know that for you, the—the human aspect of musical performance is really important. Do you have any thoughts about this?

VERCOE: Yeah. Well, yes. Marvin Minsky has developed a sense that music is perhaps a bug in the way in which humans operate. And I can't subscribe to that. I'm so much in love with music of different cultures, not necessarily the—the one culture. But, for instance, in the case of music that's come out of my own, sort of, heritage, which is basically English, I suppose—

I'm still very much in love with 16th-century choral music, of the—or some of the 15th-century c*hansons* out of Europe, some of my favorite composers, [Antoine] Busnois and [Gilles] Binchois, and people like that. I just love those composers to death. And it's—I just love hearing that. And that's—those are the kinds of pieces that I will put on at home and listen to, more so than anything that's highly experimental, which I suppose becomes a bit more—bit removed from the humanistic aspect of—of who we are.

I just find that some of the earlier music, when there's a—a mature tradition, as happened, you know, in the, as I say, in the—with the 15th-century *chansons* or the late 16th-century composers or something, middle baroque, then, of course, [J.S.] Bach, later. I do appreciate all those. But I have a rather universal view of—of music. And it doesn't have to involve technology at all to please me.

So with my friend Marcus Thompson on music faculty here, I mean, he and I are both sort of closet Renaissance choral performers. You know, we—that's what we like to listen to. So we often go to concerts that—when they're—together, to hear something, when the Tallis Singers are in town, we'll go to that, for instance.

LARSON: Do you ever sing privately, any of the two-part Renaissance stuff?

VERCOE: I always use the two-part Renaissance things, the—in my teaching of counterpoint. I still think my teaching of coun—counterpoint is my—16th century counterpoint is my favorite subject. It remains that.

LARSON: But with Marcus, have you ever sung any, kind of, stuff with him?

VERCOE: No, we haven't, actually.

LARSON: Ah, I just wondered.

VERCOE: No, no. But I'm a big supporter in the—continuing the choral tradition that came to New Zealand through the, I suppose, the English choral schools. There's a—one choral school in New Zealand. Its down in Christchurch. It's a copy of the—sort of, the King's College, Cambridge. Or maybe it was Oxford, the—the Christ College in Oxford, that has continued this real choral school, where kids go and, sort of, part of their education is, you know, singing in the—in the cathedral choir.

And I've had an experience in the last year when, 12 months ago, 13 months ago, Christchurch suffered a huge earthquake that took down the cathedral, along with most of the rest of the town. And I have come to its rescue in a way by immediately doing something to try to keep the choral tradition going, try to keep people leaving town, which is the worst thing that can happen to a city, when people pick up and move somewhere else. So I've actually created a scholarship there for choral—for—to send the—some of the members of the choristers in the cathedral choir in Christchurch to Cathedral Grammar [School].

And, sort of, that's my—that was my way of responding to the earthquake. I decided giving money to—for bricks and mortars—bricks and mortar was probably the wrong thing to do at this day and age since they're still having tremors. But doing something that will sustain the choral music in the—in that city is something that I feel very dear to. And I've taken a big part in that. I visit there quite often.

LARSON: That's really beautiful. I want to thank you so much for coming for this second interview. This has just been—been really wonderful. And you have just been so generous with your time. And—and, so I want to thank you again. And thank you, Chris, for—for coming today.

VERCOE: Good. My pleasure.

[End of Interview]