Code Together Podcast
Episode 2: Sparking Innovation through Standards
Host: Nicole Huesman, Intel
Guests: Geoff Lowney, Intel; Andrew Richards, Codeplay Software

---

Nicole Huesman: Welcome to *Code Together*, an interview series where we explore the possibilities of cross-architecture development with those who live it. I'm your host, Nicole Huesman. The oneAPI cross-industry initiative was launched about six months ago as a way to make it easier for developers to program increasingly data-intensive workloads for a diversity of hardware architectures. However, this concept is not new. So you may ask yourself, why now? What's different?

Today I'm excited to welcome two special guests to the program to give us their insights. Geoff Lowney, Senior Fellow at Intel, is one of the architects of oneAPI. Hi, Geoff!

Geoff Lowney: Hi, Nicole. Glad to be here.

Nicole Huesman: And Andrew Richards, CEO with Codeplay Software. Welcome, Andrew.

Andrew Richards: Hi, Nicole.

Nicole Huesman: Andrew, you and I had a chance to talk at Supercomputing late last year about your experiences on the path to cross-architecture computing. Can you share some of those experiences with our listeners?

Andrew Richards: Yeah. So I actually come from a video games background. So Codeplay is a company which really started as a video games tools company. So the idea is that we would produce compilers and tools to enable people to write video games and, you know, in the early days of video games, we would just get hand-me-down hardware from previous supercomputers. We ended up with these graphics processes that could do really incredible things. And, so we would produce tools for doing all the graphics and video games, and people started using them for supercomputers and artificial intelligence, and it was a bit of a surprise for us at the time.

Nicole Huesman: So there's been this shift, from video games, from HPC and video games, and then this shift to AI. Can you talk about that?

Andrew Richards: Yeah. So in a way, AI is like turning a graphics processor backwards. You know, a graphics processor was designed to take all of these, you know, representations of the world and then convert it into graphics. You're taking huge amounts of data. You're then processing it massively in parallel. And then out of the other end, you want to pop out, you know, a representation of the world. In a video game, you have a representation of the world and you convert that into something that people can see. You have AI as well. You know, I used to like these sports games. So, you know, I wrote a tennis game and you'd have to get a computer character to try and understand what was going on and play tennis with you. And in a way, modern AI is an extension of that.

Code Together Podcast
Episode 2: Sparking Innovation through Standards
Host: Nicole Huesman, Intel
Guests: Geoff Lowney, Intel; Andrew Richards, Codeplay Software

---

Nicole Huesman: And then your experience in SYCL. Right? That led into SYCL as well?

Andrew Richards: It did. Yeah. So, when I started Codeplay, we would write games in C, which seemed pretty amazingly high level. And very soon after the company started, everyone said we don't program in C anymore; we program in C++. So, one of the things that the game developers have had to do was, simultaneously achieve very high performance but at the same time, run on lots of different platforms because there were lots of different game platforms from different hardware companies, and you know, there's consoles and portable consoles and there's PCs and stuff. So they had to write code that was going to run at very high performance on all of those systems. And so early on, they started adopting C++ and really turbocharging some C++ techniques and some of those techniques have been pushing into high performance computing and also AI. And, the HPC people and the AI people have turbocharged those C++ techniques that video game developers were pioneering. Intel was also an early pioneer of C++ techniques as well, you know.

And, what became clear later on is the only way that it was gonna work was if we standardized it. The only way you can get a software developers and hardware companies working together, the only way we can create a vision where we say, actually you can write C++ and make it run very fast on lots of different platforms is by having standards. And standards allow lots of people to work together. You know, we've, we've taken these C++ ideas that were designed for, you know, different games consoles and different computers, and really run with that and made it go on really extremely different architectures. It's been fantastic to see that something that was started as just a little standards group enables such an amazing achievement really.

Nicole Huesman: Absolutely. And oneAPI really benefits from the knowledge—that well-treaded path—that has come before it, right?

Andrew Richards: It does. And you know, oneAPI leverages SYCL. SYCL leverages a whole lot of C++ techniques that came before it. So I think if you're looking at things like oneAPI and SYCL, you are looking at a whole range of developments, one after the other after the other, you know, this accumulation of knowledge and experience that's gone into these systems. That's what a really successful standards body does. It just takes all of that knowledge, all of the innovation, and goes, right, okay, can we take all of the best experience, all the best knowledge, and integrate it into something that's then standard and lots of people can use. And that's what we've done with SYCL and oneAPI.

Nicole Huesman: Excellent. And Geoff as the architect of oneAPI, can you share how Intel has worked with the community and why it's important that oneAPI is a cross-industry initiative?

Code Together Podcast
Episode 2: Sparking Innovation through Standards
Host: Nicole Huesman, Intel
Guests: Geoff Lowney, Intel; Andrew Richards, Codeplay Software

---

Geoff Lowney: As Andrew was saying, the way computers today are made of different compute units. Accelerators, GPUs. It's not just a simple CPU problem anymore. And Intel, as the major CPU company, was maybe a little late to recognize that, but over the past five years, it's become really clear that accelerators really deliver extraordinary performance. And they needed to become a standard part of everyone's compute platform. And, you know, part of that is the big AI revolution.

So on the programming side, at Intel, we're looking at this transformation and we want to deal with modern programmers. So we'd like to work with C++. And we wanted to bring data parallelism, meaning we'd be able to do millions of computations at the same time and to be able to specify that easily. And the only established solution was SYCL. It was, you know, a standard that Andrew and others had developed as part of the Khronos community. And, I looked at that and realized that this is where we want to go. And so for the past two years, Intel has really gone all in on SYCL, and we put a lot of energy into it in an open way.

And, I would say that's another thing that people may not realize about Intel. You know, we're famous for being the windows PC, which was definitely a closed ecosystem. But over the past 10 years, we have been really seeing the value of being a part of the open source movement. And somewhat independently of our move to SYCL, we have moved all our compiler infrastructure to LLVM so that we are both embracing the language standard that Andrew developed, but we're also embracing sort of the community best practice of how to develop compilers. And the community is centered around LLVM and we're contributing to that.

Nicole Huesman: That's fantastic. Thank you. Codeplay has done a lot in terms of contributing to oneAPI. Andrew, can you talk about that?

Andrew Richards: The most relevant contribution that we've made recently is to enable people to develop using oneAPI on Nvidia GPUs. So if you're developing on an Nvidia GPU today, you can check out the source code that we've committed, which takes all of the open source work that Intel did to get SYCL with DPC++ and then also their libraries as well. So then you can check out a version of that that will run on Nvidia GPUs and it will go directly down to the Nvidia-specific APIs. So it doesn't go through any intermediate layers like OpenCL or anything like that; it goes directly down to the Nvidia-specific APIs and uses Nvidia library routines. And what that means is, you get all of the performance benefits as if you'd program natively for an Nvidia GPU, but code in standard SYCL. Maybe you've used some of the Intel-specific extensions, but you know, all of that stuff is kind of aligned with the way the standard is going. And that means that you can then write your code on an Nvidia GPU, get massive performance today, and also then run on other architectures today and in the future.

Code Together Podcast
Episode 2: Sparking Innovation through Standards
Host: Nicole Huesman, Intel
Guests: Geoff Lowney, Intel; Andrew Richards, Codeplay Software

---

Geoff Lowney: And I just want to chime in. The real benefit of what we're trying to do is let the programmer write in one environment, one programming language, one set of libraries, and be able to move the program to an Nvidia GPU, an Intel CPU and Intel GPU, make it easy for data parallelism to exist broadly and move between platforms. And, you know, of course you'd do some tuning as you do that motion, but the value of a standard lets people innovate and, the value of their work is preserved, and it's something more than what runs on a single piece of hardware.

Nicole Huesman: Andrew, what's next in terms of how Codeplay envisions contributing to oneAPI?

Andrew Richards: We are working with quite a few different other hardware vendors to be able to support a variety of different platforms. So part of our business model is to sell this kind of support to different processor vendors. And that means that you can use the same source code on technology that might go into a self-driving car, for example. So you can take stuff from the research lab and then put it into production, into a self-driving car. We support platforms that will go into smart phones, for example, very low power devices. That's all part of the vision is, is to enable us to support lots of different hardware.

Another key thing about SYCL is to enable people to write their own C++ libraries that achieve high performance on lots of different platforms. And what SYCL and oneAPI do let you do is it does let you write libraries, open source libraries or closed source libraries, that do perform really well on lots of different hardware. So you should see the whole school vision as being an enabler to enable people to write these performance-portable libraries. And those libraries actually can work in the real world, delivering maximum performance on an FPGA, on a DSP, on a GPU, or a CPU. And yeah, you have to go into them and tune the libraries for different classes of algorithms. We see people doing that already. The Exascale computing program is funding lots of work in this field. We're seeing work in European HPC as well. Similar work is going on in robotics and the AI field as well, to enable people to write these C++ libraries that can handle performance portability across very different devices and really easy to use.

Nicole Huesman: Yeah. If we're really going to make this path to exascale and data-intensive workloads work, developers are really going to need something like oneAPI.

Andrew Richards: They do. And, and you know, if you want to work in this field, you want to build these libraries, having standards and standard platforms like oneAPI is a massive benefit. Otherwise, how would you do it.

Nicole Huesman: Right, right.

Code Together Podcast
Episode 2: Sparking Innovation through Standards
Host: Nicole Huesman, Intel
Guests: Geoff Lowney, Intel; Andrew Richards, Codeplay Software

---

Andrew Richards: How would you do it without having a really solid platform on an industry standard and a committee behind it? No one company controls these standards. We've enabled this now.

Nicole Huesman: Geoff, can you share with us where you see oneAPI going?

Geoff Lowney: Well, as we've been describing, we want oneAPI to become the standard interface for programming data parallel hardware. So you've got this phenomenon that there is a huge amount of innovation in data parallel software at a very high level. Also, there's a lot of innovation at the hardware level that people are architecting new types of accelerators and the world is changing pretty quickly there. And in order for both sides to innovate, you need this interface in the middle so that the software guys can innovate, the hardware guys can innovate, and there's a common waste or narrower waste that things go through. And that's what we want oneAPI to become. We want it to be standard. We want the industry to support it. We want it to be open and open source. And so that's our vision, that this becomes the standard for data parallel programming at this low level, performance level programming.

Andrew Richards: I agree with what Geoff has just said. It's, it's interesting, isn't it? That a standard which says you must do it this way actually has this counterintuitive effect of actually making it easier for people to innovate. That's the great thing about a standard. It allows processor vendors to actually innovate and design something new.

Nicole Huesman: I really love what you said earlier, Andrew, which is, it's really the great enabler.

Andrew Richards: Can I, can I talk about that? We have this challenge, which is there are a very wide variety of different processor architectures that can achieve performance on different kinds of workloads. So software developers could go off and design libraries that enable people to evaluate the different processor architectures and build complex software. And it turns out that if you go and try to build that, you'll discover that you need the standard underneath it. Like you can't build that software without having a layer to build on top of. And that's the problem that we've solved.

Nicole Huesman: Geoff, for those who want to contribute or those who want to use oneAPI, what are the best places to get started?

Geoff Lowney: If you Google oneapi.com. If you take a look there, that will give you pointers to all the specifications, all the different implementations, all the open source efforts. That's a place to let you download software to go. Now, the nice thing of what we're talking about is its portability, is that you can just start programming on your laptop, and then you can

---

take that same program and bring it to, you know, a bigger platform and see more performance.

Nicole Huesman: Geoff, thanks for joining us today. This conversation has been so much fun.

Geoff Lowney: I'm glad to have a chance. This has really been a lot of fun and I hope people enjoy programming in oneAPI; it should be a lot of fun for everybody.

Nicole Huesman: And Andrew, thanks for joining the conversation. We look forward to having you back on the program.

Andrew Richards: Yes, it was great being on the program. The website SYCL.tech—that's a really good place to start and it will link to all the oneAPI stuff and the other things going on in the SYCL group.

Nicole Huesman: Fantastic. Thank you. We really look forward to having you both back on the show. For all of you listening, thanks so much for tuning in. You can join the conversation at oneapi.com. Until next time!