
What's Most Pressing for Programmers?

(2008-11-03) - Contributed by Darryl K. Taft

A Microsoft Professional Developers Conference panel on the future of programming languages looks at what is on programmers' minds. The PDC panel of experts debates what's best for programmers and languages.

LOS ANGELES—What are some of the most pressing issues facing developers today, and what can be done with programming languages to help with them?

Those were among the questions posed to a group of language and programming experts at the Microsoft PDC (Professional Developers Conference) here.

Gilad Bracha, Anders Hejlsberg, Douglas Crockford, Wolfram Schulte and Jeremy Siek made up the distinguished panel of computer language designers and researchers addressing "The Future of Programming Languages." And the moderator was no slouch either. Erik Meijer, a Microsoft software architect and language expert in his own right, moderated the panel. Meijer was influential in the evolution of the Haskell language and is the leader of Microsoft's "Volta" project to simplify Web and cloud development.

The panel touched on a wide variety of issues, not only including identifying the most pressing issues facing developers, but also such topics as whether IDEs (integrated development environments) matter more than languages, whether modeling is important, the degree to which programmers should be allowed freedom with the language and the inevitable dynamic language versus static language debate.

First, a bit about the panelists ... Gilad Bracha is the creator of the Newspeak programming language. He is currently a distinguished engineer at Cadence Design Systems; previously he was a computational theologian and distinguished engineer at Sun Microsystems. Douglas Crockford is a senior JavaScript architect at Yahoo. He is known for his work in introducing JSON (JavaScript Object Notation). Anders Hejlsberg, a technical fellow in the Developer Division at Microsoft, is the chief designer of the C# programming language and a key participant in the development of the Microsoft .NET framework. Hejlsberg also developed Turbo Pascal, the first-ever IDE, and the Delphi language. Wolfram Schulte is a senior researcher at Microsoft, and Jeremy Siek is an assistant professor at the University of Colorado. Siek's areas of research include generic programming, programming language design and compiler optimization.

Regarding IDEs, Bracha said, "I come from a world where IDEs matter a lot. They are enormously important, but the language is also enormously important."

Hejlsberg said IDEs certainly do matter, "but a lot less than they did 25 years ago." He said frameworks and IDEs have dwarfed languages, but languages remain important. However, Hejlsberg lamented the fact that languages evolve so slowly as compared with other areas of computing.

Schulte said he believes, "languages and libraries don't matter so much. You have to look at what problem you want to solve and then pick the language." Indeed, Crockford said he encourages developers to learn as many languages as possible.

Yet, when asked whether languages should be designed by committee or by a benevolent dictator, all five panelists, in unison, replied:
"Dictator!"

Crockford discussed the evolution of ECMAScript, upon which JavaScript is based, and said although a standards body or committee may be stodgy, it is the structure the organization provides that is most important.

{mospagebreak title=Dynamic vs. Static}

Meanwhile, helping to spur the dynamic versus static debate, an attendee asked Hejlsberg why his C# language wants to be JavaScript. Hejlsberg said Microsoft is not adding dynamic capabilities to C# to force people to use a dynamic language.

"But there are many interesting things out there that you might want to talk to ... that inherently [are] not statically typed," Hejlsberg said. "And we can build a common solution for talking to these environments. I think that's super-useful."

On the question of what the most pressing issues facing developers are, Siek said he believes a key problem is, "How do we get frameworks to work with each other and how do we keep them in sync?"

Schulte said he believes the biggest problem is complexity in programming and languages. Crockford's addition to the list was "insecurity" and he said, "We need more programming languages that deal with that ... so our operating systems are designed with secure languages." Bracha said we need languages that are built for the networked world, such as the Web or the cloud, but that also are well suited to "local" work in a client or local server environment.

And, unwavering in his focus in recent years, Hejlsberg said:
"Concurrency is one of the biggest problems facing us right now. The core problems of concurrency are still there."

That they are. And there were several sessions and announcements at the PDC that attempted give developers help in this area. However, the "Future of Programming Languages" panel was pure pleasure for any programming enthusiast or developer. I only scratched the surface here. But I'll be putting more of the points the panelists made into other articles. You have to save some for later.