

# Sun's Gosling: Java Is Open Source and Doing Just Fine

May 15 2007

---

Q&A: James Gosling, the father of Java, sits down for a candid interview with eWEEK about the open-sourcing of Java.

James Gosling, the creator of the Java language, is a Sun Microsystems vice president and fellow. Gosling spoke with eWEEK senior editor Darryl K. Taft at JavaOne and discussed the open-sourcing of Java, new directions for the Java platform and programming in general, new Sun technology and competing with Microsoft.

**How did Java become another C++? It seems you created Java to enable developers to avoid the morass of C++ and now many developers say Java suffers from many of the same issues of bloat.**

Well, you can take that question in a lot of different directions. I think that Java the language has actually done a pretty good job of avoiding the rat's nest of complexities that C++ got into. And with C++, a lot of its issues were that it was based on a design from a long time ago when looking at security and the rest of that weren't really as big of a deal.

And the evolution of Java the language has been really thoroughly scrutinized and vetted by the community. So I feel pretty happy there. The part that feels like it's gotten just unbelievably complex and hairy is all of the APIs. The set of libraries available in Java is just mind numbing. It's far huger than C++ ever got to. And in some sense that's an

inevitable consequence of the range of things that people are building.

It's another one of these areas where this little thing known as the Peter Principle applies. The Peter Principle is that people get promoted to their level of incompetence. As a guy who's spent a lot of time trying to build tools to try to make developers' lives easier, if you make your life easier there are those that say: Oh, developer, you're just sitting around doing nothing, so I'll just give you more to-do items. And so if you look at the complexity of things people are building today, even compared to just five years ago it's just off the charts. So, with many of the systems people are building today, I have no idea how you would do them with other technologies.

## **Why did Sun create a fully new scripting language rather than leverage something that already existed such as Groovy? Why make developers have to learn yet another scripting language with JavaFX Script?**

Well, it's not like there's a shortage of scripting languages. There's got to be just about a billion of them out there. A number of them are fairly popular. The problem is a lot of the power from a scripting language comes from trying to fairly tightly target a particular application domain. And the majority of the scripting languages out there were really put together around generating Web pages. And that's a fundamentally different thing than building a rich graphical user interface. And JavaFX Script is all about that.

So just at like a basic level, most of these scripting languages are very transaction oriented. A request comes in; you generate a page and boom you're done. Whereas in graphical user interfaces there's a real progression of time. Things happen over time...even really subtle things like a piece of text appearing. It doesn't just appear; it slowly unfolds.

You slowly change its opacity, slowly change its size, slowly change whatever. There are a lot of things that are about animation behavior that are just different in the graphical user interface world.

So Chris - Oliver, the Sun engineer who conceived JavaFX Script - looked at this stuff in a fairly different way, and came up with something that works in that universe pretty well.

If you tried to take JavaFX Script and apply it to generating Web pages, it would be awful. But that's not what it was made for.

**So he [Chris Oliver] did it on his own time? Was it a project at Sun?**

He started working on it when he was at SeeBeyond and then we bought SeeBeyond. And we thought that it was pretty cool so we kept him working on it after the acquisition. It got to the point where we wanted people to see it and play with it and kick the tires. Chris is a little intimidated by the whole "getting his baby out there" thing, but...

**On somewhat that same theme, what sort of language do you think will emerge for Web 2.0 and mashup programming?**

I don't know. The thing that I find disturbing about programming around Web 2.0 right now is it's become really confused and messed up. There are like 10,000 ways to do anything. And the cobbling together of this and that and the other thing is sort of like pouring paint together. If you pour a bunch of buckets of paint together you pretty much always get a really ugly brown. And it's difficult to come up with a coherent architecture approach when you do things that way.

At some level, whatever Web 2.0 evolves to be, it is what it is. I hate the term because nobody knows what it means. But the core technology that's used all over the place is JavaScript. And it's sort of unfortunate where it has ended up, because it's kind of the definition of fractured.

People criticized us about being really strict about testing, etc., around Java. And for us one of the big counter examples for why that was important was the way JavaScript has gone. When you look at average pieces of JavaScript code, they - programs - spend a small fraction of the time doing what they want to do and the rest of it is "if I'm on Internet Explorer version this then I have to do it that way, and if I'm on Safari this or Opera that or Firefox this other thing then I have to do it another way."

And then it sort of ratchets up a level where there are all these libraries that try to unify all the differences. Then there are like libraries of libraries and it becomes deeply meta-circular. And really all of these pieces of complexity are all just about trying to deal with the fact that the platform is screwed up.

## **But won't we get into that with JavaFX Script?**

Well, there is no JavaScript there. JavaFX Script is a language that fundamentally compiles to the Java VM - virtual machine - . I mean it's not like a whole new language and whole new APIs and all that. It's a really interesting way to drive Swing and Java2D.

So all of the rendering libraries that you get with the Java platform ... and the ability to animate and all the other stuff that platform offers is all there. All that stuff is just not present in JavaScript.

## **Are you happy with the open-source direction of Sun**

**and Java? Word was early on you sort of pushed back on it. Graham Hamilton - a former lead architect for Java at Sun - is said to have quit over it. Are you ok with the route the company has taken?**

Oh, I'm really happy with the way things have gone. We have been running this as an almost open-source project since the beginning. Our big issue was around interoperability and consistency. And that was my one nervousness about the open-source world is that the way many open-source things end up going is that they just fracture.

And so many of these things end up with 27,000 different versions for no particularly good reason.

Also, the license that we had been using was pretty much the Mozilla license with the testing requirement. But it feels like we've gotten to a point where the developer community is strong enough and feels strongly enough about consistency and interoperability and quality that market pressures will keep things on track. And those that try to misbehave will have issues with the market.

**When the talk began to heat up about Sun's plans to possibly open-source Java, I did a column that said Sun plans to open-source Java and I don't care...**

And for the vast majority of folks the open-sourcing thing is not a big deal at all. In fact we had more people who were negatively concerned about open-sourcing. Probably our single biggest problem with open-sourcing was not freaking out the other half of the world.

Because one of the things that people had gotten to really value about

Java was all the consistency and interoperability and testing. The fact that you have an institution like NASDAQ running Java at its core....

**Do you think, regarding the testing issue, that Apache has a legitimate claim in wanting unfettered access to the TCK - test compatibility kit - without "field of use" restrictions?**

I don't exactly understand the Apache gripe. You might want to talk to Simon - Phipps, Sun's chief open source officer - about that. One of the issues we've had with open-sourcing is that often people try to generalize the open-source community as this one big kumbaya happy family, but in fact it's a bunch of warring states. They all have their Great Wall of China and they lob stones back and forth.

**That's a perfect description because there are real wars going on.**

There are real wars. And it's really hard to be friendly to the open-source community because if you're friendly to this camp then you're viewed as an enemy by that camp. And one of the things we got stuck on was that we really like the Apache folks, but we also rather liked the GPL guys. And the Apache folks were very angry at us for picking GPL. But we had to pick something. If we'd picked the Apache license, the GPL crowd would be upset with us.

*Copyright 2007 by Ziff Davis Media, Distributed by United Press International*

Citation: Sun's Gosling: Java Is Open Source and Doing Just Fine (2007, May 15) retrieved 1 May 2024 from <https://phys.org/news/2007-05-sun-gosling-java-source-fine.html>

This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.