An Interview with SunSoft Chief Scientist Rob Gingell: Part Two

One of the most popular presentations at the recent Solaris Developer Conference was given by Rob Gingell — Chief Scientist at SunSoft™. Since so many attendees were interested in Rob's perspective on the future of the Solaris operating environment, and his detailed description of the Solaris API, the Catalyst Flash interviewed Rob for the benefit of all developers.

Rob has been with Sun Microsystems since 1985, and with SunSoft since its inception in 1991. He came to SunSoft to work on the Solaris operating environment, and has been responsible for a number of the key technologies in SunOS™ and Solaris; in particular the development of the ABI and its enabling capability of dynamic linking. The following interview gives an excellent insight into the internals of the Solaris operating environment and Rob's vision for SunSoft products.

In Part 1 of this interview (Fall, 1993), Rob Gingell explained SunSoft's strategic views of the Solaris API. The conclusion of this two-part interview picks up with the discussion of SunSoft's strategy to deliver next-generation software validation technology.

You've classified the components of the Solaris API in the context of three key issues: availability, stability, and validation. The next step in API validation is source-level client-server validation. What is that?

This is where you can verify that the server of an interface, such as Solaris, delivers that interface properly from release to release. We also want to verify that developers use it properly.

The one limitation to source interfaces is that validation can only test static relationships between applications and systems software. This leads to what we regard as the next generation of validation technology — the binary client-server. This new technology provides not only static validation, but dynamic validation during the developer's SQA process to determine whether they are using the product correctly. Dynamic validation is something that we haven't been able to do before, and is one of the reasons why we did

Solaris 2 in the first place. If you go back to 1987, our early discussions and announcements centered around the construction of an Application Binary Interface (ABI) where a true "shrinkwrapped" relationship could exist between Solaris, applications, and hardware products. This capability was enabled through technologies that we created back then in the form of dynamic linking technology, which has since become pervasive in the computing industry. With these changes, SunSoft is trying to minimize testing for developers to only when they change their code. At the moment, they have to test when anything — the systems software or the application — changes. And of course the trouble is, applications and systems software are changing more and more rapidly. Even if software development stopped, hardware platform developers would continue to release new platforms at a rate of at least one per year. So even if software remains stable, application developers still have to deal with changes resulting from the evolving hardware, which is going to happen faster, rather than slower.

Your earlier analogy does bring up the question of how customers adapt to the constant change in the computer industry.

Yes, well the trouble is that most people believe that when interfaces are defined they are cast in concrete and won't be subject to change. But the fact is, customers expect change — remember... they're the ones who request new features in the first place. They expect change because they realize the potential of computing and are anxious for new technology. So the reality is that there will always be continuous innovation.

Let's go back to the ABI issue for a minute. Could you describe the SPARC Compliance Definition?

Sure. Let's review a few examples of things that exist in Solaris and discuss where they live (see Figure 1). We have the SPARC Compliance Definition (SCD) or ABI, and we have the Solaris operating environment, which now has more than 1.4 million units in total volume. The SCD, which lives on SPARC only, is somewhat smaller, and the ABI volume, of which the SCD is a superset would be just over one million.

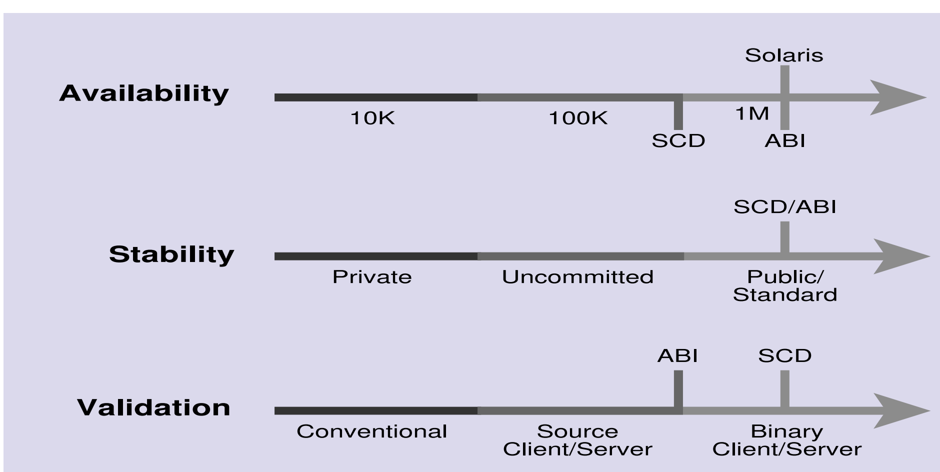

Figure 1. SPARC Compliance Definition (SCD) ABI

In terms of stability, the SCD and ABI live in the public and standard domain, and hence can't change in an incompatible fashion unless the whole industry agrees upon it. The most important thing is that SCD is the first demonstration of a validation technology that allows us to inspect applications and client systems to validate whether or not they behave correctly with respect to each other.

I'm sure our readers are curious about the Solaris Binary Interfaces. What can you tell us about them?

We have a set of building blocks that constitute Solaris (see Figure 2). Even if software developers don't rely on this building block approach, SunSoft will still use it in order to build a more robust product that won't break every time it comes to market. We want to create this same relationship between the applications that ISVs and customers create, and between the systems and peripheral hardware vendors.

Developers want an open systems marketplace where they can publish their products and know they can sit on the shelf without becoming invalidated by future versions of systems product, and vice-versa. Once this happens, then we can begin talking about new ways of delivering systems software while remaining innovative and responsive to what the customers need. One of the things that SunSoft has demonstrated in the past are new channel marketplace systems that allow developers to get applications and systems software components to market much quicker and more efficiently. Programs like Catalyst℠ CDware™ enable us to ship applications through our channels directly to customers, while providing developers with an effective way of reaching Solaris users.

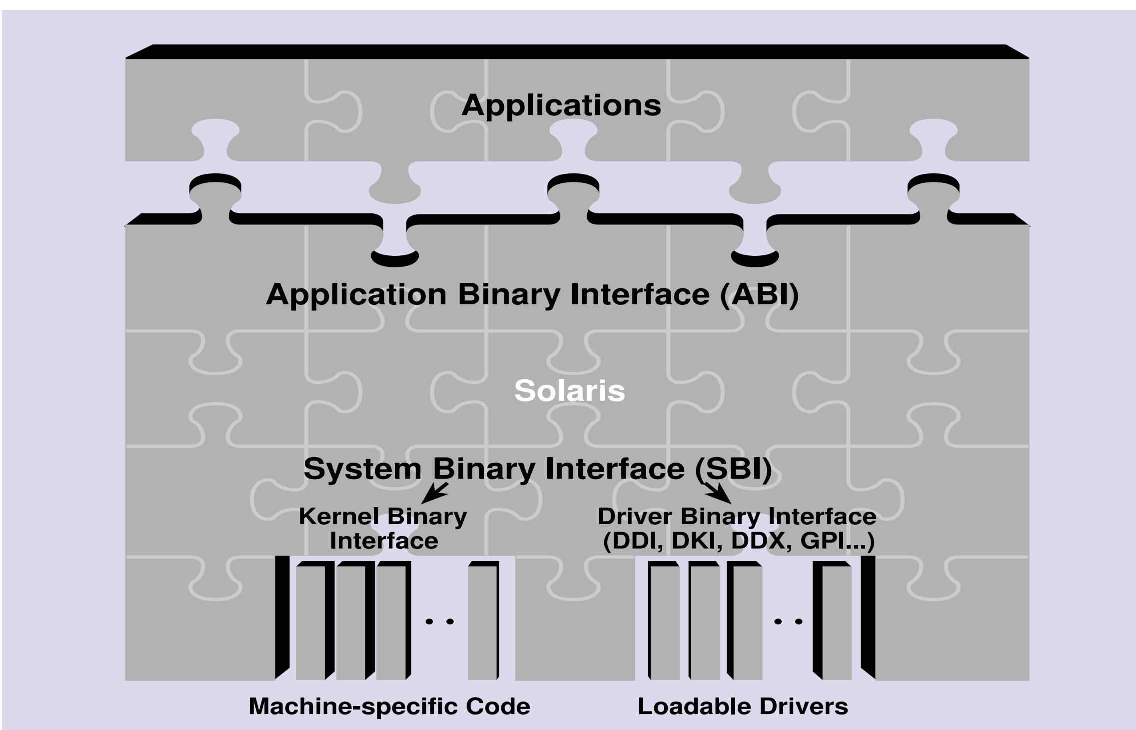

Figure 2. Solaris Binary Interface

It's also just as important as the individual applications however, customers want one piece of media that supports all possible hardware they use. Systems administration is the most significant cost to customers in terms of running open systems environments. One way to make systems administrators more efficient, and thus able to support more systems, is to ship them one piece of media that supports all of the hardware they have. This gives us the opportunity to create a channel to our customers and enables our partners to reach them, while introducing new qualities into that channel that even the computer megastores do not provide.

Technology cycles tend to bring about changes which beget other changes. One of the changes we want to bring about at SunSoft is how we service our customers so that they can get products on a more rapid, more customized, and more cost effective basis than they are able to do today.

Delivery of product to customers is an important topic for developers to be thinking about...

Definitely. The component level binary-model testing approach will enable us to change the delivery paradigm in the future such that we can create highly reliable channels, so that our customers can have the same expectation that they have for other consumer products. For example, wouldn't it be great if customers thought computing was as reliable as the telephone? In order to do this, we have to change the cost structure and overcome the need to test all of our products together at the same time, because the rate of change occurring in the industry makes it prohibitive.

When you examine how we build software today, it is analogous to how things were done in the 1830s. There has been no Eli Whitney or assembly lines in the software industry. We are still in the “artisan stage” of software. What we need in order to survive is a pluralistic industry. This could create a software industrial revolution allowing repeatable events and interchangeable parts which customers know they can put together themselves without requiring a massive staff to do so. Over the next few product cycles, we will change how developers do development. In part, we are going to try to help the SQA process to evolve, so that engineers can develop things that they know will work when they are done.

I like your analogy of the software industry being in its “artisan” stage...

Thank you! (laughing) Part of what you will see coming from SunSoft in the future is an advanced set of tools that allow your application to run while you are developing it, and have it dynamically report to you where your boundaries are. As a result, you'll be able to discover what your costs are, and make an informed decision about whether you want to pay the costs, take the risks, and reap the advantages. The primary benefit to developers is the ability to move to quality assurance on their terms, instead of ours. We believe this can dramatically increase efficiency. For example, we estimate that if we had been in a position to instantly implement this, we could have saved $35 million on the SQAing of Solaris 2.0 and 2.1.

We are almost out of time, Rob. Do you have any final thoughts to share with our readers?

Sure. I'd like to close by saying that there are a lot of capabilities in the Solaris operating environment, delivered in a highly integrated product. The capabilities are broad in terms of their stability, their efficacy, and their validation systems.

SunSoft is spending a great deal of effort laying the groundwork necessary to dramatically change the cost that developers pay in doing business with these technologies. The principal reason that SunSoft developed Solaris 2, and our reason for asking developers to make this investment with us, is to set the technological stage that enables these validation systems. It is always difficult to make an investment where the return comes later. We believe that a developer's investment return will more than justify itself. We want to look at this as if they were signing up with one of the new hot start-ups in terms of the return that this is going to provide for them. Their commitment will help us all get started on the “rich get richer” cycle of the computer industry of volume, applications, customers and efficiency.

That was very helpful. Thanks for taking the time to share your views with us.

My pleasure.