

W A S A T C H
DIGITAL

iQ

Connecting Utah's Business and Technology Community

July 2002
Volume 2
Issue 7
u.s. \$4.95
¥ 620

Web Services

Bond. Phil Bond.

eBay CEO Bids Big

Siebel Says
This Is The Place



perspectives

Craig Burton,
JanusLogix

by Bill Kerig

Craig Burton is in a Big Dogs T-shirt, shorts and sandals, hair askew, Einstein style. He leads me to a shaded area of his Sandy home where I sit under a canopy of trees. The setting is pleasant; wicker furniture on an elevated porch. This is not where I'd expect to encounter radical plans to alter the future of computing, but that indeed is what I'm about to hear.

Burton takes his place in the shade and I reflect that this is a man who casts a long shadow. One of the founders of Novell, Burton developed such concepts as file server technology, hardware independence, fault tolerance, Universal NetWare Architecture, NetWare Open Systems, metadirectory, and The Network Services Model. His concepts have accelerated the movement toward transparent multi-vendor computing.

The logo on the T-shirt is prescient, non-coincidental; in the world of visionary technologists Burton is a big dog.

Technological advancement is an inside-outside process, a herky-jerky evolution plotted on a mobius strip. Sometimes the breakthroughs are visible, usable, and appreciated from the outside. Sometimes the big changes happen on the far edge of the strip and you only find out about them later, after they've rocked the world. Listen to Burton and it's easy to believe the computing universe is about to shift dramatically, that we are in a period of Copernican upheaval. The geniuses are working on the far side of the strip. Remembering that Copernicus died of a brain hemorrhage I tell Burton to feel free to dumb down his explanations.

"It's not about dumbing down," he said "It's about taking something that I've spent 20 years thinking about and making it clear enough to describe in 20 minutes. We're creating stuff that's never been thought about, or done, there's no metaphor, no analogy, no lexicon! That makes it very hard."

"So you won't need the whole 20 minutes, then?"

He laughs, and I remember that Galileo was just a guitar player, Newton just an orphaned farmer. The tape recorder spins.

Wasatch Digital iQ: Better start from the beginning.

Craig Burton: In relative terms of the history of computing, we're just barely coming out of the Dark Ages. There's a lot of invention happening right now, and Web services are the next step. Web services are like the calculus. You know what the calculus did?

DiQ: Mostly made it hard to graduate from high school.

CB: Right. The *de riguer*, or the mindset in Newton's time, was to model the universe as this big metastructure that defined how everything worked. Everything was systematic, unchanging. Newton and Leibniz went the other way. They kept breaking things in half until they got to a point where they could calculate and mathematically represent objects that were changing dynamically.

That changed every notion of science. Everything else was built on the calculus—atomic theory, relative space and time theory—everything came out of that because they inverted the thinking. That's what we're talking about here. We're creating the calculus of the Internet.

DiQ: Usually big advancements happen in response to a problem. What's the problem with the computing industry?

CB: It moves too slow. The Internet collapse is a direct result of the fact that things couldn't move fast enough. You have all these huge infrastructures and no interoperability. On the Internet, a generational cycle is 90 days. Every 90 days another thing comes out, but it takes 10 years or more for the core stuff to change. It's just plodding along because of the politics.

DiQ: What's an example of something that's been held back.

CB: Before the Internet, the functionality of closed email systems was much better than what we have in the current Internet email system. You could track when an email happened. When they opened it. If they hadn't opened it, you could go get it and take it back. You could track and filter and do all this stuff that you still can't do because there isn't interoperability between systems.

DiQ: But we've got universal connectivity.

CB: Yes, but at the same time, the functionality was retracted. We still don't have the basic functionality in email that we had ten years ago because the infrastructures can't talk to each other.

DiQ: Not to keep digressing, but now would be a good time for an infrastructure primer.

CB: What's infrastructure? It's like air, water, food—the basic elements. The low-level stuff that you need to live. That's infrastructure.

Software infrastructure is what allows solutions to function. Solutions get built on infrastructure. We used to have things like Lotus Notes, which was the transport, the services, the infrastructure and the applications all in one big package. A monolithic system. If you wanted to add a new service to Notes, you'd have to find one of the three guys on the planet who could do it. It would take two years and it would break everything else.

Microsoft, Sun and everyone else build, these big frameworks that are all-inclusive, and all-exclusive of anyone else's. They take the whole package and own it. That's a meta approach.

DiQ: And they're what's been holding back the industry?

CB: Yes. Think about how long it takes for a multi-celled creature to evolve. A long time, right? So they're not a very good genetic study group. People who study genetics study fruit flies. They are born, have sex, regenerate and die in a day. That's nice. We need to move from the mammal rate of evolution to the fruit fly rate.

DiQ: So with Web services we speed up the process of the evolution of software. Why now?

CB: The commercialization of the browser forced the industry to agree on a transport protocol. That was good. As a computing industry, we've never had political agreement.

DiQ: Now there is an agreed-upon transport standard, XML.

CB: And SOAP, WSDL, and UDDI. In the last year, the industry has agreed on a set of technologies that enables basic interoperability at the services level. This is the beginning of component-based interoperability that we've been trying to get to for a long time. It's a very important event.

DiQ: Now all the disparate systems out there will work together?

CB: We've agreed on transport. To some degree we've agreed on the next layer, which is system application services—things like HTTP, LDAP, SMTP. But note that they're all named by protocols. This is where things are still broken. The de rigueur is to use static protocols to define services, to force everybody to speak the same language.

DiQ: To make everybody speak Microsoft or Sun?

CB: Yes, and that's ridiculous. That's what we're going to shift. JanusLogix technology is a fundamentally new infrastructure to change the mindset of how to do services. We call it "Janiculum." We're changing the approach of how infrastructure is built.

With the JanusLogix system, you can use Microsoft infrastructure, Sun, or the independent developer's. Janiculum takes a nano approach as opposed to a meta approach to providing infrastructure and creating software solutions. We build extremely small components that are inclusive of everyone else's systems. Janiculum doesn't exclude anyone's infrastructure.

DiQ: So the idea behind the whole thing is to move all the functions of today's O/S onto the Internet?

CB: You're jumping ahead, but that's what this is going to do. The O/S on your desktop is going to become a commodity. And the services that you run will happen there on the Internet as opposed to on your computer.

DiQ: And real Web services are not happening today because of interoperability problems?

CB: "Web services" is a poor term because it's confusing. For the last 20 years, the industry has been trying to figure out infrastructure so that you can use some sort of re-useable code. It's been moving toward this notion of components for a long time.

We've had this history of re-useable code. CORBA was a framework for re-usable code. Distributed Computing Environment, DCP, was another one that died a horrible death. Hundreds of millions of dollars invested in these and they died. Microsoft built the Component Object Model (COM), and that died too. Microsoft.Net is going to replace that. It's their big framework for building reusable software.

The problem is that everybody builds this big framework and there's no integration, no interoperability. You've got to buy into the Microsoft solution to make it work. It's the supplier pushing that solution on you. We need to get out of way and let solutions happen dynamically.

Fruit flies . . . are born, have sex, regenerate and die in a day. We need to move from the mammal rate of evolution to the fruit fly rate.

DiQ: eBay, for example just couples the buyer and seller, gets out of the way, and let's them work out the payment mechanism.

CB: That's good. What we want is to let computing systems automatically adjust to the way they interact with each other in the same way that people do on eBay.

Networking works on a request/response model. I make a request and you respond. The request is very formal, specific, in a certain format. The response is also very formal. A service like that takes a very long time to build because you have to have the requester and the responder always match protocols.

DiQ: So you're not looking for agreement on one protocol, but rather to make the protocol arbitrary. How?

CB: By componentizing and breaking down the stream and then matching it to the responder or requestor.

DiQ: A translation engine, a semantics engine?

CB: More of a protocol engine. There's no magic to it. You still have to build the pieces to it that describe the translation.

DiQ: Gotta ask about the name. Who's Janus?

CB: Janus is the Roman god of doors and entries. He has two heads, one looking to future and one to the past.

DiQ: And the future is Web services.

CB: Web services gives us an opening to address the problem of how to build dynamic protocols. What we do is define, use, execute and finish with a service on the fly.

DiQ: On the fruit fly. . .

CB: Glad you caught the pun. You're still awake. So the excitement about Web services is that it allows independent software vendors to start competing with Microsoft and Sun and IBM.

DiQ: Why would they agree to this?

CB: I think it happened sort of by accident. XML came out of a really smart group of independent people. In general, programmers are pretty apolitical. On occasion, they get together and do something before management can figure out what's happening. There's such a gap between technologists and management that occasionally this sort of thing happens.

DiQ: Do you have the system already built?

CB: Yes, we've got a prototype working.

DiQ: What are some specific advantages of your product?

CB: For example, Caldera has a Linux-based operating system management platform, but they can't manage Windows computers. We can take the Linux component and map it to the management client on Windows and make each think they're talking to the right service. We take the XML and map it in real time. We just get right in the middle of it. It's redirection.

At Novell, when your computer wanted to open a file we'd hijack the event and send it to NetWare and let NetWare open the file. NetWare made the operating system independent of what the wire looked like. Now we're making services independent of protocol. It's just NetWare trickery at the next level up.

DiQ: What do you mean by "the next level up?"

CB: For example, we can increase Outlook's functional capability by adding new services. Let's suppose you want to do some market tracking of your customer. You want to know whether they clicked on the link, whether they opened the email, whether they downloaded the attachment. You can use your existing email system and get additional functionality without buying a big service.

DiQ: Who's this for?

CB: The customer, the reseller—whoever can make a business out of it. That's the other rule of ubiquity. Let someone else make money off your infrastructure.

DiQ: So, how do you make money?

CB: We sell the basic infrastructure. We sell a server, just like I did with NetWare. Pay me \$2,500 bucks and go do your thing. One shot.

I want people with real problems to be able to go out and buy it. It's a simple business model, and I know it works. Right now, the concept is a little tough to get your arms around, but NetWare was just as tough.

DiQ: And this is going to catalyze a second wave of Internet prosperity?

CB: Second wave is a good way to look at it. As these things get solved, the way computing works will completely shift. When that happens, it will have dramatic political, cultural and individual impact. 