Jeremy Allison is a lead developer of the Samba Project, which exists to remove barriers of interoperability with Microsoft Windows. He has been employed by a number of different companies to work on Samba, including Sun Microsystems, SGI, Cygnus Solutions, VA Systems and others. He is currently employed by HP, lives in San Jose, California, and works in Cupertino. Here, he speaks to Daniel James in his capacity as a Samba project member

The interoperability dance

Opening Windows to a wider world

What's your remit at HP now?

I officially work for the Linux and Open Source Lab based in Colorado, although I'm in the Bay Area. I spend probably about 25 per cent of my time on the Open Source Review Board, which checks when HP puts open source into any product, or releases anything as open source. It goes through a review process where we check it for legality, make sure we're doing the licence and everything. The other time I get to fix Samba bugs, and travel - which is alright, for a while... I just came back from Shanghai, Hong Kong, and I think I was in Germany, but I can't remember why. All the hotels look the same, but you know when you're in Europe because you can't get broadband. Even Shanghai had broadband, for heavens sake!

A roving ambassador for Samba?

Kind of, and I get to talk about what HP is doing. HP actually does quite a bit, and they don't get much credit for it. I don't quite know why that is. IBM seems to have a very high profile; they do a lot of stuff, but HP does a tremendous amount of grunt work. The thing I like most about what HP does is that it doesn't have its own licence - there's no 'HP Open Source Licence'. I work with the best lawyer in the world - the HP lawyer completely understands the licences, and as he's put it to me, 'There's nothing you can't do with a combination of GPL, LGPL, MIT or BSD. Any business objective you can achieve with those.' You don't need the HP Public Licence 1.7 or whatever.

Why do you think corporations do tend to devise their own licences?

Because they can! The legal department says 'We've got to protect our interests', so they write something, which when you look at it is exactly the same, with slightly different wording, to seven other licences. The IBM Public Licence is a pain in the arse, basically.

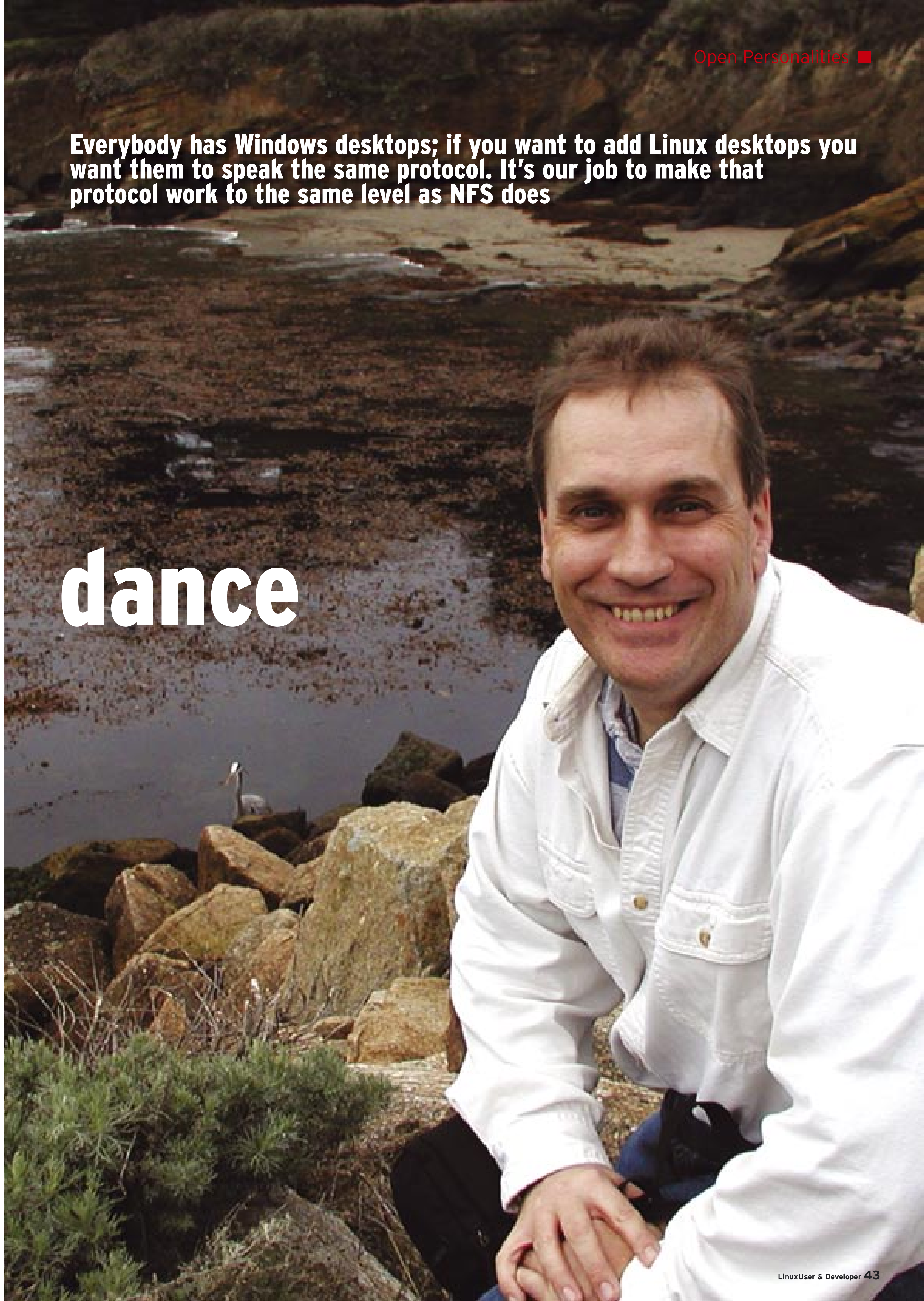
A diversity of licences doesn't help when it comes to maintenance?

It's just confusing. If I want to combine code from the Mozilla Public Licence, the Firebird Public Licence, the Apple Public Licence, the CA and IBM licences, can I do that? God knows! Licence fragmentation is as bad a thing as UNIX fragmentation was, and that sucked.

Would you advocate that those organisations reconsider?

Yes - you don't have to do it all GPL; there are many things that you can do with a combination of those licences. For instance, Mozilla might as well just be LGPL, because that's what it really is anyway. I think the Apple licence is pretty close to BSD, except with a nasty patent poison pill. You're not going to need that - just pick a standard one!

HP do actually get the open source thing, and not just on an engineering level, but on a management level - and that's very unusual.



Everybody has Windows desktops; if you want to add Linux desktops you want them to speak the same protocol. It's our job to make that protocol work to the same level as NFS does

The open source companies like Red Hat and SUSE got it from a fundamental level because that's where they came from, but for a non-open-source-start-up company to get it is really rare. I would say that Sun and IBM haven't got it - they adopt open source but they haven't got it.

They haven't internalised it?

Because they're still: 'we've got to keep control, we have to have our own licences'. The power of it is essentially: knowing when to give up control. If you want to keep control, just keep the damn thing proprietary; then you've got all the control you'll ever want. If you're going to do open source, don't do a half-measure - do it properly.

Is it due to the culture of HP, which was originally a start-up with two guys in a garage?

It's interesting - I ran into some of the old HP people, fifteen years ago. Everybody said 'HP is a really special place to work'. When you run into those people, you understand why - they're just a different breed.

A technically-led company, rather than a management-led one?

I would say so - a company where people were incredibly important.

Samba 2.2 sites are coming out of maintenance - do you have an estimate for the number of sites running 2.2?

Not at all. Most of the bug reports and questions we get are on 3.x. There are still people out there running 1.9.17 - if it works for them and they don't want to change it, that's fine. But at a certain point you have to start saying 'we can't keep backporting ever-changing fixes to earlier versions'. Now, that's not to say that if someone else really wants to keep that in maintenance they can't do that. The code is all open and the changes are pretty trivial; with any security fixes we do, you can see what they are and the patches are fully described. In terms of spending our resources, they're better spent concentrating on the current release.

Is there a sort of natural selection, in that if people don't maintain a version it will die away?

This is open source - if someone wants to maintain that, they can. It's just that the Samba team has to concentrate on what we're doing in the future, moving things forward. It's not like that code is abandoned - it's as useful as people want it to be. If the 2.2 code is working for people and they're in a secure environment and they don't care, why do they need to change? If something works, there is absolutely no reason for people to upgrade.

If it's not in a web-facing environment, for example?

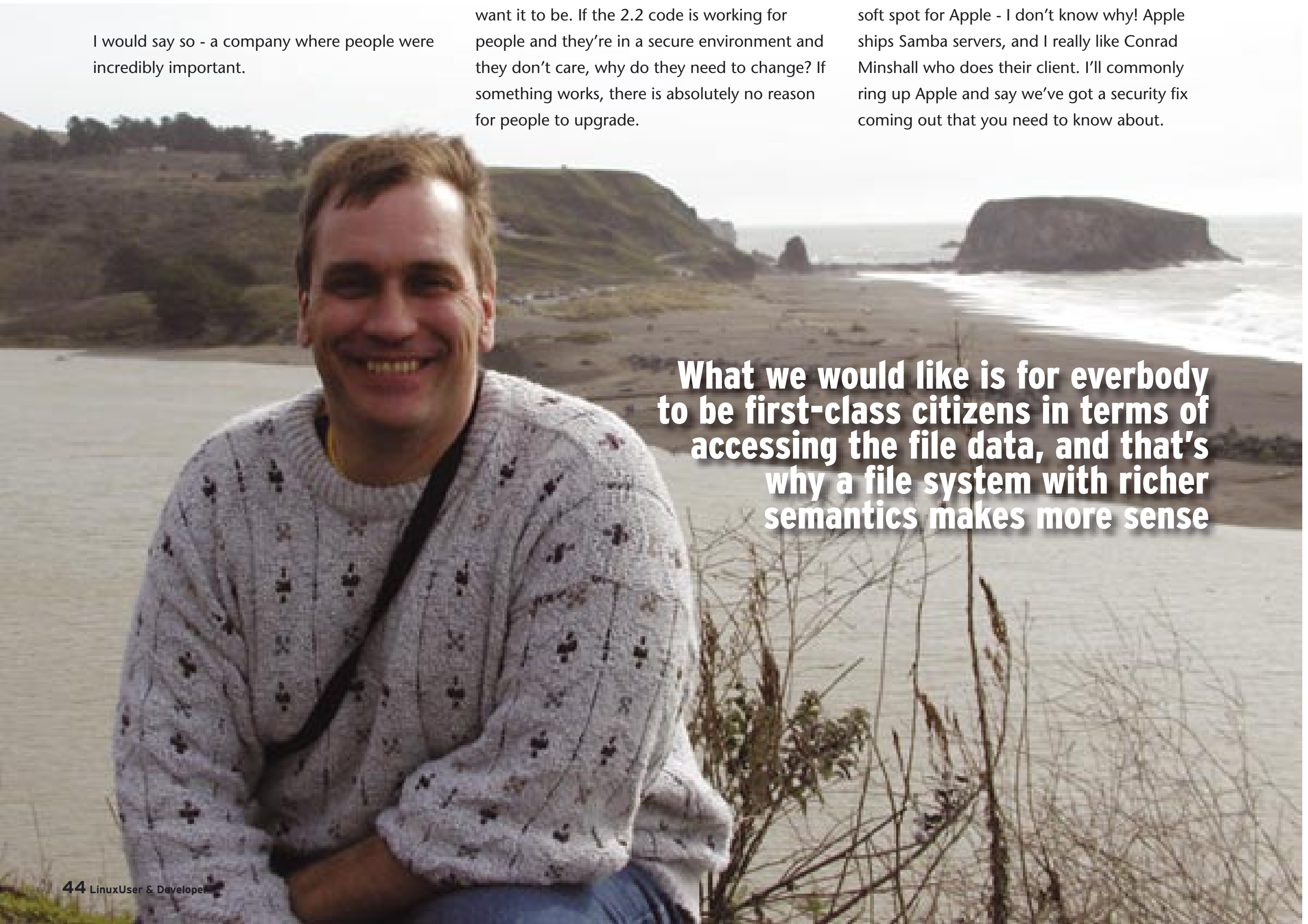
If it's not in an external-facing environment, and you're happy with your network, then you don't need to worry about security all that much. A lot of organisations are like that.

Could there be a problem if third parties are packaging Samba as part of some bigger solution, and people aren't even aware that they're running an older version?

The third parties who package Samba - the ones who are competent and interested in maintaining their products - have a relationship with us already, so they know. We do get people coming to us and saying: 'What was that security fix you just put in 3.0? Because we want to put that in our version.' The good vendors already do that.

Which vendors do you have the closest relationship with?

I don't want to pick any favourites, but HP and IBM employ a lot of Samba team members. We have very close relationships with them. I have a soft spot for Apple - I don't know why! Apple ships Samba servers, and I really like Conrad Minshall who does their client. I'll commonly ring up Apple and say we've got a security fix coming out that you need to know about.



What we would like is for everbody to be first-class citizens in terms of accessing the file data, and that's why a file system with richer semantics makes more sense

That's running on OS X and serving to Windows?

Serving Windows and also Mac, because Mac clients now use SMB natively because of Conrad's work. The UNIX to UNIX aspects of that are something that I'm working with him and Steve French from IBM to make work.

All the hotels look the same, but you know when you're in Europe because you can't get broadband

Do you think it will finish off NFS?

Well, no - NFS and NFS v4 will continue to be active in the datacentre and as a sound backend filesystem. But to the desktop it's a non-starter. Everybody has Windows desktops; if you want to add Linux desktops you want them to speak the same protocol. It's our job to make that protocol work to the same level as NFS does.

We're actually better in terms of correct UNIX filesystem semantics - we have better semantics than NFS does. Because of the oplocks, we provide better performance; the Linux clients can safely cache more of the file. They just take an oplock out on it - they can cache the entire file locally, write to it and flush it back, just like Windows clients do.

Can that work with any filesystem on the Linux server side? How much will Samba 4 plug into features of the native filesystem?

We don't know because Samba 4 isn't finished, and 'Tridg' (Andrew Tridgell) is doing most of the work on the filesystem side. I have to keep maintaining Samba 3. As I put it to him: we have to keep maintaining Samba 3 so that people care when Samba 4 comes out!

I don't know what the Linux filesystem space will look like when Samba 4 ships; my guess is that it won't be much different from what it is now. The main thing will still be the POSIX backend.

It's agreed that some form of journaling filesystem is essential...

We already have those. It's the extra metadata - the issues with interoperating with Windows are that Windows has boatloads of extra metadata. Historically it hasn't been used. The danger is that Microsoft now no longer ships or maintains any servers or clients that don't support NTFS with all the metadata. What that means is that Microsoft applications will increasingly start to depend on that metadata being there.

Both Linux filesystems and NTFS are becoming more complex...

We had a big argument on the Linux filesystems mailing list recently. I stuck my oar in, saying: 'These are the things we need for Samba. They might be insane, but...' Linus did accept that. There's a Solaris interface that could be extremely useful for us called 'open at' - it allows you to open a file, and then open a different, named stream within that file as though it were a directory entry within it. ReiserFS has one way of doing that, and Hans Reiser was saying: 'Have a completely different API - make all files directories'. Linus said 'no - that will suck - it will break applications.' So I said: 'why not use 'open at?' because there's already a precedent for it - let's just adopt the same interface and try and keep some standards.'

I know Linus is happy with that, but I'm not sure about the rest of the kernel community. Generally, what Linus is happy with is what gets implemented. I'm hoping that will win, and at that point at least we can make the extra name stream stuff work on both Solaris and Linux.

There's nothing you can't do with a combination of GPL, LGPL, MIT or BSD. Any business objective you can achieve with those licenses

Hopefully then other UNIX's like AIX and HP-UX will follow and adopt the same interface.

Is Samba a layer of abstraction for any filesystem that's underneath it?

Yes, but some filesystems are better than others. You can expose more functionality when you have a filesystem with more functionality. You can always fake it by storing things in crappy little dot-files or whatever, but when you do that what happens is the native UNIX tools become second-class citizens. Doing a move means you've lost metadata, so we don't want that.

What we would like is for everbody to be first-class citizens in terms of accessing the file data, and that's why a file system with richer semantics makes more sense. Reiser 4 is probably the first one that will have that, but I'm more interested in making sure the interface to it is clean. That's why I like 'open at' - it might not be the best interface in the world, but it's already there and people are already using it.

Is the rich metadata in NTFS fully capable of being supported by Reiser 4 and other forthcoming filesystems?

Reiser 4 definitely; they need some extensions to the other filesystems to support it. I'm more interested in the kernel interface to it - I don't care how they implement it underneath, as long as it's fast! Reiser 4 is there now, and the others will probably follow.

So we'll end up with a choice of filesystems again?

Yes - right now, probably XFS is the best one to support Samba because it has extended attributes and the Apple support, and it's very fast. A lot of the appliance vendors ship using XFS on Linux.

The other interesting things coming up are Lustre and the other object-based filesystems. Right now they're only targetting POSIX, but what will they target in future? The joker in the pack is Novell - what will they do when they port their NetWare filesystem to Linux? That could be our perfect filesystem - we don't know yet, because we don't know what it will look like. The Samba team has had conference calls with Novell engineers, essentially to make sure that the kernel interface for that will be right and we can get access to the features inside Novell's NetWare filesystem for Linux.

Have Novell indicated what licence that will be under?

No clue - that's for them to decide. I would love it to be GPL and in the standard kernel; I could understand if they feel that this is their crown jewels and they don't want to give it away.

Is there anything else you wanted to say about Samba?

The Samba 4 work is where a lot of the exciting stuff is going on; unfortunately, that's not the area I'm working in! As my mother often said: after the Lord Mayor's Parade, which is Samba 4, comes the muck cart - which is Samba 3! I really like Samba 4 - I use it as a test suite against Samba 3. I will probably start getting more involved when Samba 4 is a little more complete.

If the 2.2 code is working for people and they're in a secure environment and they don't care, why do they need to change? If something works, there is absolutely no reason for people to upgrade

The main thing I want to make sure is that people have a seamless upgrade experience. I don't mind if some things break, because it is a major release; it is a major revision. I'm not going to say that every single little crappy parameter you've ever had in Samba 3 will continue to work. But the major things have to, so people have to be able to take a working Samba 3 configuration, run it through some kind of migration script, and then have a working Samba 4 configuration.

Of course, we never support downgrades! As I like to say, 'save your configuration before doing an upgrade...' If you don't, then that's your lookout. It has to be seamless.

There are a lot of interesting things coming along in Samba 4. It has a full DCE/RPC mechanism over TCP as well as SMB; there are ways to plug-in back-end DCE pipes into Samba 4. It will be a complete implementation of essentially any generic RPC mechanism running on top of UNIX. You'll be able to write IDL, compile it and plug in back-end pipes. Any service that is done over generic DCE/RPC, coming in the portmapper on port 135, we will be able to do in Samba 4 - that is already working.

Somebody's already working on a DCOM implementation - God knows why, but I can see the code being checked in to Samba 4. I think it's one of those 'because I can' things - it's a very nice infrastructure, and my guess is that it will start becoming stable when someone wrestles it out of Andrew's hands and says 'Stop making changes! It's as elegant as it's every going to be!' The thing you've got to remember about working with Andrew is that he's basically a genius, and so it's hard to keep up.

Maintaining Samba 3, you have to match what Microsoft does, bug for bug...

Pretty much, although obviously the data corruption and filesystem bugs, or crash bugs - we don't match those. We have our own crash bugs!

Do you have to follow security issues with Windows?

Not really, because our implementation is just completely different. The times we have to care are when they make a change for security, we have to look at how it affects us. So if they say you've got to have signing now, that will force us into doing signing, which we have already.

We're at the stage now where we understand what they do well enough that we are hitting some interesting bounds. For instance, we now have code in Samba 4 that tests what happens if you're in the middle of scanning a directory, you make an arbitrary change and then you move back. When do you see the change? Do you see it immediately, or do you have to close the directory handler and re-open? It's that level of detail that you really need to understand if you're going to have a perfect implementation.

To give an example of something I've changed recently to fix an issue that people have had for a while with Excel files: It turns out that there's a feature in SMB whereby a client will open a file, get an oplock on it, then attempt to open the file again, breaking its own oplock; deliberately do an open which will get a share mode violation. If you fail it immediately, as you should do, the application fails. It turns out that what happens in Windows is that you have a one second delay - when you get a share mode violation, you don't return it for one second. You have to keep processing incoming packets from the client, in the hope that somewhere in that one second you will get a close for the original open that will then allow the second open to succeed.

Is that quite possibly an ugly hack?

Possibly. In a multi-user environment, Excel depended on that to work. If you didn't do that, then very occasionally people would get 'the file is open by another user' when it wasn't. I think it's not so much an ugly hack, as a side effect of the way their redirect and server worked together that the Excel coders didn't notice. So they were depending on doing that without realising that they were depending on it, because they're at a level way above the SMB redirector.

But it's that level of detail that we have to get right, and this is why I had a talk - I haven't given it for a while now, because most people have given up on doing their own - saying 'Why writing an SMB server is hard'. In all the published specs, that's not mentioned at all. If you followed the published specs, you would just get it wrong, and Excel would fail. The knowledge of how that stuff works that is embedded in Samba is what makes Samba so valuable as a piece of code.

So you're drawing your own map as you go?

Yes - my guess is that Microsoft don't have regression tools that are as good as those we have in Samba 4. I'm not 100 per cent sure, but we have found differences between Windows 2000 and Windows 2003. What I'd like them to do is adopt the Samba 4 test tools and maybe donate some changes back. It would be lovely if Microsoft would participate with us in making these test tools really tie down the semantics, as I would love to co-operate with them on that. I think it will help them with regressions in the future too, to make sure that nothing broke in future versions of Windows.

I suppose the underlying principle of Samba is that Microsoft isn't going away, so we have to accommodate it.

I would love to see Microsoft become IBM; they are the same size, and they have that much money. They're in the same position that IBM was. Whenever I talk to Microsoft people, what I say is 'Why are you not shipping SQL Server and Exchange on Linux? It's the fastest growing server OS: sell licences, make money. Why are you not doing this?' None of them have a really good answer. I think it would show that they'd matured to the point that they accept there's going to be a mixed environment.

They do have Services for UNIX...

They've made that available for free, and I think Samba is a big part of why they made that free. The trouble is that it isn't very good - it's an NFS server, and it's not a great one. Anyone wanting to run Windows clients is much better putting Samba servers in; for mixed Windows and Linux clients it's better to put Samba in and use Steve French's CIFS client inside Linux.