Peter Buneman Speaks Out

on Phylogeny, the Integration of Databases and Programming Languages, Curated Databases, British Plumbing, the Value of Talking to Users, When to Ignore the Literature, and More

by Marianne Winslett

http://homepages.inf.ed.ac.uk/opb/

[A brief reminder: http://www.sigmod.org/interviews has video versions of many installments of this column, as well as many interviews that have not yet appeared in print. For example, the video version of this issue's column has been available for years.]

Welcome to this installment of ACM SIGMOD Record's series of interviews with distinguished members of the database community. I'm Marianne Winslett, and today we are at the 2006 SIGMOD conference in Chicago. I have here with me Peter Buneman, who is a professor in the School of Informatics at the University of Edinburgh. Before that, he was a professor at the University of Pennsylvania for many years. He has research interests in databases, programming languages, scientific databases, and mathematical phylogeny. Peter and his coauthors received a best paper award from Computer Networks in 2002. He is an ACM Fellow, a trustee of the VLDB Endowment, and a Fellow of the Royal Society of Edinburgh. His PhD is from the University of Warwick. So, Peter, welcome!

I understand that you like very good whiskey, and Scotland is famous for its whiskey. Is that why you went back to Scotland?

Well, I do like very good whiskey, although I do not drink it very much; but no, that is not the reason I went back to Scotland. It is a rather complicated story. I have always had an affection for Scotland; I know the place very well. We have had a house in a remote part of Scotland for a long time. My return was certainly helped by the fact that I was very comfortable in Edinburgh, which is a city that I know very well, and where I have lots of friends. I probably wouldn't have tried anything much more adventurous than going back to Edinburgh. I think the reasons for going back will emerge later on as we talk.

You have been building a very strong research group in Edinburgh. I know what a big job that is! What motivates you to take on that task?

I think I went slightly mad! [Laughs.] I have to say that I was extremely comfortable and happy at the University of Pennsylvania. I had wonderful colleagues there. But there is a point at which you realize you are becoming a bit of an old fogy, and part of the furniture. At that point you should step aside and let your younger colleagues take charge – they are superbly competent, so why not? – and you should try something else. Either you are going to stay there forever, or you should get out and do something new. Edinburgh made me a very attractive offer, and I thought it would be an interesting challenge to build up a really good database research group in the UK, which frankly has lacked one for a long time. We have been quite successful in building up a database group. Also, as I am sure you are aware, the UK has also hired Georg Gottlob, so I think that all combined, we are doing very well.

You are not the only database researcher who has recently moved from the US to Europe. Have conditions recently changed in some way that makes this route more attractive?

I think Europe is becoming a bit more competitive now, waking up to the need to bring computer science researchers back from the US. There are increasing funds and opportunities for doing this, and I was attracted partly on such funding. I don't think it is a major trend at the moment. I don't think we are going to see a mass defection.

What is the difference between the educational systems at Edinburgh and the University of Pennsylvania – or, if you prefer, between the UK and the US?

From the point of view of a teacher, I think they are rather minor, until you get to the PhD. The PhD in the UK is much shorter; the UK students are expected to do it in three years. As you know, in databases in the US, the PhD can take 5 or 6 years, sometimes even longer. That means that when UK PhDs graduate, their publication records aren't typically what you would expect of a US PhD. This means that in order to be competitive in the US market, they have to go on and do postdocs. So there are many more postdoctoral positions in the UK than in the US, and this is partially how we have populated our database group at Edinburgh.

I see this trend toward postdocs increasing in the US. I think as the academic job market saturates here in the US, computer science researchers generally are going to be in less immediate demand. Then you are going to see an increasing number of postdocs in universities. Postdocs are already common in other subjects, such as physics and biology, where students hardly ever go straight from a PhD into a teaching job.

You began your research career in functional programming and type theory. How did you get from there to here?

Actually, I began my research career in mathematical phylogeny. Perhaps I should tell the story slightly differently. I started by doing some work on the phylogeny – the derivation – of medieval manuscripts that used to be copied from each other. A professor of the English language wanted to know what the tree of copying was. There was some very interesting mathematics to be done in the work to compute that tree. When we did that work, we knew that the techniques were also applicable in genetics, but there wasn't much genetic material around at that time to try it out on. We tried it on some common protein sequences and it worked very well.

Then, several things happened. One of them was that I met my wife, who is an American, and we decided to move to America. When I got to America, I told the chairman of my department about what I had been doing. He said that medieval manuscripts weren't exactly wealth-creating, and that I should go off and talk to people who do databases. So that is how I got into the field. Of course now, mathematical phylogeny is big business, and the techniques I worked on are quite widely used.

I was always fascinated by programming languages, and I always felt there should be a stronger connection between programming languages and databases. I used to enjoy programming very much, functional programming in particular.

You were one of the first people to work in the intersection of databases and programming languages. Why is it that after 25 years of people thinking about the relationship between programming languages and databases, the two communities largely still don't talk to each other?

I don't think that is quite true. I think there is quite a lot of back and forth between the two communities. I know a number of database people going to programming language conferences, and vice versa. For example, certainly there are at least two talks at this conference (SIGMOD 2006) given by programming language people. And I think that it is quite good that the database community brings these people in. If you look at PODS, you will find a lot of papers now with people worrying seriously about type systems and things like that. That work is directly influenced by the programming language community.

I think the more interesting question is whether there has been any *practical* meeting of the two communities; and I think there has been. There is now, of course, the realization that SQL is a functional programming language and people are trying to keep it that way, and make it compositional, and talk about these kinds of properties. But more importantly, if you look at the things that are going on in C# and Linq, you will see exactly what we have been striving for: a clean integration of databases into nice programming languages. And a lot of the ideas lead back, through a chain of work, to the original work on the relationship of programming languages to databases.

The problem is that in computer science, we expect immediate gratification. We don't expect an invention in physics to have commercial impact – though if it did, that would be regarded as great. Even in computer science, sometimes it can take a while for a good new idea to have commercial impact.

You are a coauthor of a well-known paper that talks about the principles of object-oriented databases. Mike Stonebraker recently told us that "object databases are a zero billion dollar market." So was all that object-oriented database research a waste of time?

I don't think so. First of all, that paper was not on object-oriented databases, it was on *type systems* for databases. It was called "Types and Persistence in Database Programming Languages" (ACM Computing Surveys, June 1987). And when we wrote it, we did not know a lot about object-oriented databases. But, I think generally, that work, and the work on object-oriented databases, although they have not been hugely successful commercially, have been very influential in other aspects of databases. For example, we have object-relational systems, and we have Mike Stonebraker's Postgres, all of which were heavily influenced by research in object-oriented databases. I don't know whether object databases are a zero billion dollar market or not, but the work was influential, and it certainly changed the course of database research. Again, we should not necessarily judge ideas by their immediate commercial value.

Mike Stonebraker also said that XML is a very good format for transferring data, but other than that, it is largely a pain. Do you agree?

I have to agree! XML is designed for data exchange. As such, it is very nice; it is great to have a standard format for moving data around, and not have everyone invent a new one, which is what was happening before. But I think that what has happened is that people are trying to turn XML into a data model, or something like that. Look, I am a teacher; I have to teach about databases for a living. Now, as you know, when you teach relational database design, it is quite difficult. Students have a hard time doing it, and they create really bad designs. XML is a serialization format, and things like XML Schema and DTDs are very complicated. To get these students to design databases using XML Schema would be terribly difficult, really impossible. We shouldn't be thinking that way. It would complicate the process unnecessarily to say that we should also be thinking about the serialization of data when we design its schema. That would just be wrong. I think the basic idea of XML is very nice, but all this elaborate superstructure has gone too far. When it comes to database design, we have to simplify. We have to look for very simple core ideas, and not try to make it more and more complicated.

You have been working on data provenance for a number of years. Where will the research on data provenance have its impact?

I hope ultimately it will have impact in various areas. One is the way in which we design databases and use them. For example, it is only recently that we have started to think that maybe when we use databases we should actually make them record their whole history. It is true that in commercial databases you can roll back to a limited extent, but that is not the same as recording all history. You can't ask questions across time with a database that just supports rollback.

Just the other day, I was talking to someone who had built a database for the government to record the real estates sales in Edinburgh. As soon as a house was sold, they updated the old record to reflect the new owners. The database had absolutely no concept of history, and so the real estate lawyers were no better off than before; they had to shuffle through all the paper transactions to determine the history of a house. Omitting historical records from the database design was a huge mistake, but it is the sort of thing that is all too easily done when people design a new database.

That example points to one way in which we can see provenance potentially have an effect: we can design databases in a different way. There are people thinking about this, and devising ways

in which this notion of history can be built into database systems so that they can automatically keep provenance information properly. Remembering history is very important, because provenance is one of the major aspects of data quality, especially in scientific data. People just don't believe what they find on the internet unless they know who put it there.

How would you address the scalability issues if support for provenance is built into a typical DBMS?

That is what we are working on at the moment: we are trying to find efficient ways of recording provenance.

There is another answer to that same question, in that provenance tends to be more important in curated databases, which typically are rather small. In large databases, things are much more uniform, with large blobs of information all coming from the same place and sharing the same provenance. So, we have to find efficient coding techniques for encoding provenance in these larger databases, but I think it is going to work. In other words, I think we are going to get large databases that look fairly uniform in respect to provenance, and small databases that tend to be more heterogeneous with respect to provenance. I think we are going to find good ways of recording both kinds of provenance.

What is the next big thing in research related to data provenance?

Provenance is part of data quality, as I said earlier. Provenance is also very closely related to data integration. We tend to lose provenance information during integration. But then there are other things like annotation, database archiving, recording the history of databases, citation of data in databases – all of these are closely related to provenance, and we are just starting to work on them. And of course there is lots more work to do on data quality.

What are curated databases?

That's a good question! Go to the reference shelves of your local library; you probably haven't been there for ten years, right? A lot of old encyclopedias, dictionaries, gazetteers – people just stopped printing them around 1990 because they have all become databases. You don't go and look things up in the paper version of the Encyclopedia Britannica now; you look them up in the online subscription version of the Encyclopedia Britannica. All these books that used to be on the reference shelves of the library are being published in curated databases now.

These databases are the result of human effort: people decide which data to put into them, in an extremely labor intensive process. I just found out that the Oxford English Dictionary has about 50 people working full time putting entries into it, and another 100 people working part time.

At the same time, the ease of publishing data in this form has produced a wealth of new curated databases. This is very clear in molecular biology, where there are something like 800 curated databases. All of these were constructed by people; they are like large collaborative books. The move to curated databases has occurred in the past ten years, and we have largely ignored the problems of these people.

What are the problems of these people?

We have talked about some of them already. For example, provenance! There used to be standards that people used to explain exactly where each piece of data came from; now people are

just cutting and pasting data into their database. They don't want to ignore the problem of recording where they took the data from, but life is too short for them to record it. Provenance information is largely made up of annotations, and how do you annotate something that is in a database with a fairly rigid structure?

Of course, the knowledge in the curated database evolves, and you want to keep a history of the previous states of knowledge, so you also need to support archiving.

Database curators also face the more general problem of data quality. Errors in the data are common. Can we use automated data cleaning techniques? There is a lot of good work to be done there.

As always, there is a certain amount of data transformation going on when data is added to a curated database, and there is data integration as well.

The main point to make is that we've been thinking of provenance or lineage, or whatever it is called, in the context of views. But curated databases are not views. They are built by people, by the sweat of their brow. They are very expensive databases. I have found a lot of very interesting problems associated with curated databases, and the database community is just starting to look at them.

What is a good methodology for choosing research topics?

I know I am more associated with theory than with systems, but I have always found that the greatest inspiration comes from talking to the people who actually use databases, the customers. And here you have to be careful. If you go to a scientist and you say, "I do databases," they will come to you for help with their SQL, or with building their database, and so on. But if you dig a little deeper, you often find that they are trying to do something with their data that we haven't thought about. Most of my recent work has been prompted by doing just that – talking to users who are consumers of database technology, who are faced with these problems.

To give a very recent example, a colleague of mine at Edinburgh was doing something very strange. I couldn't figure out why he was doing this. He had one of these curated databases built by contributions from several hundred people. He was trying to give everybody credit for their bits of the database. So he was trying to produce a citation system for his database. That comes back to the question of how we can cite data in databases. That problem is something you can actually work on rather easily, and you can get some simple solutions.

How do you like to spend your time outside of work?

I spend a lot of time at my house in the highlands. I also like to go sailing or boating, and I have a boat on the sea. I also enjoy building things. I have quite a big workshop, and I make things. I think it is very relaxing. To sit at a lathe and turn a bowl – it takes your mind off everything. I also enjoy the usual things like walking and music.

What have you made in your workshop?

Last week I finished making a desk for myself, and now I can put all my computers in it. My wife hates computers with lots of cables all over the place, so I built a desk in which all the cables and wires are carefully hidden.

And I understand that you did this with equipment that you imported from America?

My wife liked the idea of building a house in Scotland, but she wasn't going to give up American plumbing or her hairdryer. I said that we could buy a hairdryer in Britain. She said no to that, so I agreed that we could buy a transformer to convert the current. Then we realized that we would be bringing a lot of other electrical items from America, and so we wanted the house to have both American and British wiring. We put mostly US plumbing in it as well. I also built a workshop, and since we had a 110 volt power supply, I loaded the moving van up with lots of power tools. At that time, and probably still now, power tools were a lot cheaper in the US than they are in the UK.

You have been the program chair of SIGMOD, PODS and ICDT. Do you have any comments on the conference system?

This is a very difficult problem, and I am not sure I have anything new to add to the discussion. It is really sad when you see good papers rejected because of incompetent refereeing, and this does happen.

There are good things and bad things about SIGMOD and VLDB. One is that they are very extensive, and inclusive; they tend not to be narrow. But everybody will tell you that their best submitted papers were rejected from these conferences, and their weaker papers were accepted. When you get older, you regard these outcomes as a statistical thing, but it is very hard to tell your students that the paper they have put their hearts into has been rejected. I wish we could somehow strengthen the quality of refereeing by ensuring that people with the appropriate skills are assigned to referee each paper.

Maybe the issues with our conference system will just shake out in time, and we will become more like other disciplines, with a more stable view of what the subject is. I think that our current problems are a necessary side effect of the nice property that our field is permanently in a state of flux, and permanently rather popular and successful as a research area.

Do you have any words of advice for fledgling or midcareer database researchers or practitioners?

When I was very young – when I was a post-doc, actually – I was reading the literature on some topic or other, when my postdoctoral advisor, who was a very brilliant but rather crusty gentleman, came in and said, "You have done enough reading! Ignore the literature. Go away and think!" It turned out to be quite good advice because two of the ideas I had were quite original, and the third one was Quicksort! Fortunately I realized that my idea for Quicksort was not original before I tried to publish it.

This was of course a very long time ago. Still, I think at some point it is a good idea to stop reading the papers, and do what I said earlier: talk to the users of databases, or talk to people who work in other fields. You often get very good ideas from just casual remarks that people make, or very odd things that they do.

Among all your past research, do you have a favorite piece of work?

I still have some fondness for the work I did in mathematical phylogeny, because it did have real impact. Those were the days when you could do very simple things, and they would work out well.

I think the cumulative effect of the work on the interaction between programming languages and databases is important, but I cannot pinpoint one particular paper. I think it has had a beneficial effect on the community.

If you magically had enough extra time to do one additional thing at work that you are not doing now, what would it be?

At work, I would love to go off and do something wacky like study the confluence of information theory and physics. I always have tremendous respect for people who have time to do that sort of thing; unfortunately, I am not likely to have the time. In databases, there are a lot of new things that I would like to keep up with that I can't. For example, I think some of this new work on probabilistic databases is very interesting. I would love to keep up with that, but I haven't been able to. But really, if I had the opportunity, I would go off and work on something completely different.

If you could change one thing about yourself as a computer science researcher, what would it be?

I would organize my time better. I am not a good organizer, and as a result I waste an awful lot of time. I think if I organized my time better I would have much more fun doing research. I am already having lots of fun!

Thank you very much for talking with me today.

Thank you.