David Maier Speaks Out

on the Impact of Object Databases, Why People Don't Use Object-Relational Features, What DB Theoreticians Should Be Doing, How to Make a Success of Yourself at a Small Institution, and More

by Marianne Winslett

David Maier http://www.cse.ogi.edu/~maier

Welcome to this installment of the ACM SIGMOD Record series of interviews with distinguished members of the database community. I'm Marianne Winslett and today we're in Madison, Wisconsin, site of the 2002 PODS and SIGMOD conference. I have here with me today David Maier, who is a professor in the Department of Computer Science and Engineering at the OGI School of Science & Engineering, formerly the Oregon Graduate Institute, at the Oregon Health & Science University. He is an ACM Fellow and an advisor to the DARPA funding agency, and he is well known for his work on database theory and object-oriented databases, including serving as an advisor for the Gemstone, ObjectStore, and O2 object database products. Recently he has been quite active in the area of scientific databases and [scientific] data management. David did his PhD work at Princeton in the 1970s and was a professor at Stony Brook before joining the Oregon Graduate Institute in the 1980s. So Dave, welcome.

Thank you.

Dave, 10 or 15 years ago, object-oriented databases were looking like the next big thing, Where did they go, or as some might put it, why did they fail?

That is an interesting question, and it's [an important question for] me, having based so much of my career on object databases. I think part of it was that [the object database companies] were targeting very niche markets. So many of the object database companies were going after [the]

computer-aided design [market]---which, in its entirety is maybe a 50-100 [million]-dollar [business]---at most---compared to billions [of dollars] in the business data applications [market]. On the other hand, there are still some very significant applications using object-oriented databases. There [are] Gemstone financial applications that run billions of dollars a month in trades. Objectivity is being used by CERN for particle physics data. In terms of where the technology has gone, [some] of it has shown up in object [relational] databases, [and] a lot of it seems very usable for XML [databases].

[I] don't see any end application users using the [new object-relational] features.

So, [object databases] may have vanished from the radar scene, but the products are still out there and still selling.

Well, [some] of the companies have gone away or been absorbed. O2 was bought by Unidata, which was bought by Informix, which was bought by IBM, and so it's sort of hard to see where [O2's technology has gone---]there might be little traces of the code someplace. Some of the companies have relabeled themselves as XML managers, so ObjectStore is mostly known for its Execlon product now. Other [object database companies] have tried to be middleware engines, integration engines.

I've heard that object-relational databases have not been the big hit that the vendors were expecting. Why is that, and what are the long-term prospects [for object relational databases]?

I think one of the reasons that [object-relational databases] are not the big hit is [that] having all of these new features in [the ORDBMS] has expanded the [schema] design space, and there's not a good design theory the way there was [with] normal form relations and so forth for relational databases. I spent a little time when I was on sabbatical in Wisconsin with Raghu Ramakrishnan looking at [the question of] what [is the schema] design space for the very simple [new object relational schema] features, and it looks like [the new features] at least [double] the number of [design] choices you have in every instance. It turns out that object modeling techniques like UML aren't [very well] suited to the object-relational models, because [the] object models that the object-relational [databases] provide don't quite match up [with what UML offers]. In terms of the future, [I] don't see any end application users using the [new object-relational] features. That might be wrong, but I get up in front of rooms [full of people] and say that, and nobody has come up and told me I'm wrong in about five years. Where [the new features] do seem to get used is for people who build these [DBMS] extensions, like extenders or cartridges or Informix's DataBlades. [Some] of those object relational features go into producing the cartridges, and in turn people buy the cartridges, so I guess they're using some of these features but in an encapsulated way.

Is there more work that the research community needs to do [on object-relational databases], or is [the needed work] more on the product end [of things]?

I think the interesting thing might be to look [at] design tools for the [object-relational schema] design space. I think people are not going to use the [new object-relational] features until they know how to map their problems to [object-relational schemas]. I don't think [object-relational

databases have] a performance issue at the moment. That may [come] in the future, once people know how to use the [new] features. [I] know for sure that there are going to be some interesting optimization problems with some of the [new] features.

[So] right now, [are schema design tools] an area that theoreticians should be looking at, or [are the issues there best explored by] experimenting with users?

Well, the group in the database community that might be the most interested [is] the modeling people. You know, the kind of person who would work on ER model tools and things like that. If [the modeling community] could come up with some good [tools] that worked like the entity relationship [modeling approach] worked with the relational model, I think would be the most useful thing [for increasing the usage of the new object-relational features].

Are there any other results from object-oriented database research that you would single out as having had long-term impact?

I think a lot of the work on [object database] architectures [has had a big impact.]

I think a lot of the work on [object database] architectures [has had a big impact---]how do you divide up the system of a database between a server and a client? [It] used to be that the database ran on one [computer] and the application ran on the other. And now if you look [at] these object-oriented systems, some of the database actually runs on the client, which is a good thing because that could be where most of the [computing] cycles are in total. [Exploring] those architectures [is] important [even for] relational systems, [so] that they might put more processing on the client side.

The other thing [that] I think will have impact is [that] I think we finally figured out after ten years how to optimize OQL, to do cost-based [query] optimization [for OQL] and solve some of the hard [optimization] problems. And I think that will [be] useful for XML query languages.

I've heard other people say [that] object databases failed because no one ever figured out how to optimize the queries. I [think] David DeWitt [said] that. So, [was that comment] true then, but no longer true?

Yeah, if we could have done in two years what we did in ten, it might have made a big difference. One of the problems was that you really need a very specialized group to do a good [query] optimizer, and most of these object database companies just weren't big enough to afford that. [S]o it took proportionally longer in academia to figure out a lot of these [query optimization] issues.

Do you think now that we know how to [optimize OQL], it will find it way into [the object database] products?

I'm not sure it will find its way there. An interesting place it might get into [is] Enterprise JavaBeans. [EJB is a] middleware architecture[. I] was just reading the [EJB] 2.0 spec, and they have object schemas and query languages now as part of that [specification]. So EJB is

looking more and more like an object database. And so maybe some of that query processing technology will find it way [into the optimization of] EJB queries. [And] I think the best place to apply it is to the XML query language.

Do you think XML is going to fare better in the marketplace than object databases?

> I think we finally figured out after ten years how to optimize OQL… if we could have done in two years what we did in ten, it might have made a big difference [in the acceptance of object databases.]

I think it already has. [People] don't realize how much [XML has] penetrated the marketplace[.] I've had people say to me, "Well, I went out there and crawled the web for XML documents and I didn't see them." [But] people have to realize that the web isn't the Internet. There's a lot of stuff that happens on the Internet that's not just putting up web pages. And it could be that you're seeing XML, but it's been translated into HTML before it hits your browser.
[In my] latest incarnation of my object database course, I had students doing databases for annotations for biological sequences that were available out of what are called bioDAS servers, [where DAS stands for] Distributed Annotation System. [With the database systems my students built,] you send in a query and you get back XML. [The XML is] meant to go directly into some kind of specialized browser[. So the XML] is out there if you know where to look for it.

[XML] is also being used a lot for messaging.

What should the role of database theory research be today?

I think [database theoreticians] should [be] trying to look at actual classes of problems and abstract away [the] hard bits [of the problems,] or the commonalities [between them. Then they should be looking at the question of] how should we go about solving those problems, or at least thinking about the problems. I don't think the role [of database theoreticians] is to keep building on yesterday's theory.

So where should they be looking [to find] these new hard problems?

I think streaming data is a great place to look[.] There's just a lot of theory and semantic work that needs to be done there.

I think something [else] we've ignored for just way too long [is] data management for the kind of data structures that scientists actually use[---]time series, multi-dimensional arrays, finite element grids. [Many theoreticians' attitudes have been that only if] it looks like a set of records, if it has the same schema as my checkbook, I'll study it. [T]here's been some interesting work, for example, on optimizing array queries[.] I think that kind of [research is needed] where you go look at the data, go find out what data structures some scientist is actually [using,] and try to look at those and see what you can figure out for them.

[You've] done a lot of work recently on databases for scientists. [That's] a niche topic in the grand scheme of things because no company's ever gotten rich selling databases to scientists. Do you think more people should be looking at issues in that domain or do we have it well enough covered that the researchers should stick to the mainstream applications, which have [a] huge market and [a] potential huge impact for [researchers' results]?

So, I'll correct one of your presuppositions. People do make money selling *databases* to scientists, just not *database management systems*. So [databases] that have been populated, [e.g.], gene-expression data, are sold.

I think [that there are] a lot of commonalities with what [scientists] want [in a database] and with what domains that have a lot of money want. [For example], time series [data] shows up in financial [applications] a lot, shows up in health and medical records [applications]. The pharmaceutical companies are very interested in databases of gene-sequence data, gene expression data, [which] many scientists are [also] interested in. So I think it's worth working on features to support those [kinds of data].

It is [economically] hard to have a new kind of database management system just for scientists. [Easy] extensions that will handle [the scientists'] kind of structures[,] hybrids of [a] traditional database management system plus some specialized file storage that's integrated [into the database management system, seem to be] what a lot of people end up doing anyway[. It] seems like a fairly credible approach, especially since a lot of scientific data is write-once.

> I don't think the role [of database theoreticians] is to keep building on yesterday's theory… I think streaming data is a great place to look [for interesting new theoretical problems.]

Of my first set of interviews for the Distinguished Profiles in Databases [interview] series, you're the only person who is not at a big-name university or research lab. How has that impacted your life?

I guess part of it is that […] if I were to have gone to MIT or [some similar school,] I would have gotten a lot of stature […] inherited from [my position] at the university[. Here] it's sort of the other way around[---]it's sort of my department or school [that] gains stature as I've succeeded and as the faculty there succeed. So you have to […] have a lot of trust in your own abilities […] to succeed in the environment that you're in, and that [your success] will bring stature to the place you are, rather than assuming [that] by just getting [a position at your university] that people will think you're great.

Do you have any words of advice for young database researchers who want to have a big impact even though they're not at a top thirty research university?

There's some measure by which [OGI is] in the top thirty[, but] maybe not the top twenty yet. I think what you have to do is maybe think a little bit about what your influence is going to be. And so for me, thinking back, a lot of my legacy is going to have been building this department that I was in. It was very small when I got there and it's getting more and more [well] known.

Another thing to realize is that you need to concentrate on a few areas[, as] a departmental strategy. Don't try to cover everything; try to get an area or two [where your researchers] achieve national recognition. [W]e've come out ranked up to 5th in the US News and World Report rankings of database programs, so [don't] think you're […] limited by the overall status of the department. [P]ick a niche or an area and emphasize that.

And also you need to exploit your unique situation. There may be things around you that aren't available to researchers elsewhere. So in my case, what was really wonderful was that Gemstone was starting in downtown Portland, and I had a long-term consulting relationship with them. And I don't think if I'd stayed at Stony Brook I would have had that kind of interaction with industry.

The other thing you can do is, of course, joint projects. I've done a lot of work with people [elsewhere---]a little bit with people at Stanford, a lot with people at Wisconsin and Brown. [Collaboration] with good people will help as well.

Now that OGI has merged with a large health sciences institution, does that mean you have to work on bioinformatics for the rest of your time there?

Right, I'm basically a database administrator now for the bioinformaticians. Not really! [laughs]

[Many database theoreticians' attitudes have been that if] it looks like a set of records, if it has the same schema as my checkbook, I'll study it.

It doesn't mean I have to [just work on bioinformatics]. There will be a lot of people that will continue doing the same work [as before,] but [the merger] presents some interesting opportunities. There [are] good groups in […] genomics and neurology and heart development, and so […] starting to build bridges and doing joint research and joint proposals is kind of fun. So I'm not forced [towards bioinformatics,] but I think I may have some opportunities that are not there for a lot of other [database researchers.] I plan to go after a number of [those opportunities] and in fact, I was already working with people in the medical informatics division there before we even were talking about merging with them.

I see. So it sounds like [the bioinformatics people are] ready for a collaboration on their end.

They are. [A] lot of the biologists suddenly find themselves thrust into this computational area. [Especially] with gene sequencing, gene expression, they have volumes of data that they've never had to deal with before. And so at least on the data management and informatics end they're really looking for help. They're happy to talk to us.

Do you have any other words of advice for fledging or mid-career database researchers or practitioners, wherever they may be?

I think my main advice is “be incomparable” [---]and realize there [are] a couple of ways to be incomparable. One [way] is to pick such a narrow area that nobody is going to be more expert than you at that area, but you run the danger that nobody else cares [about that area]. So another way to be incomparable is to have knowledge about a range of topics, such that there's no other person that knows more […] than you [about] every topic.

[T]here’s an interesting trend [that] in the last 20 years, the percentage of [tenured] faculty at institutions [that offer tenure] has dropped from about

[In] my case, I think [my incomparability comes from] doing […] both theory and systems. One of the nicest things someone ever said to me was [when] Dave DeWitt said, "Boy, you know a lot about systems for a theory person." And I think there may be theory people who say, "Boy, you know a lot about theory for a systems person." […] I would like to think I'm incomparable, [and] I think some of my influence comes from being […] conversant and having published in theory areas but also knowing a fair amount about systems.

[Knowing] about […] some areas of databases and […] some related application areas is also a good strategy [for achieving incomparability].

[…] I also work on both sides of the fence. But now that you mention it, I can't really reel off a lot of names of people who do [both theory and systems in the database community]. […] Isn't that a bad thing?

I think there are people who do [both theory and systems, but] sequentially. [M]ost mathematicians produce their great results probably by the time they’re 25 or 30. And so maybe people try to transition […], like I sort of did, from [theory to systems]. But you're right, there's not too many people who for a long time dwell on both sides [of the fence]. I think it would be good if there were more.

Many people think the system of higher education in the United States won't continue as we now know it, due to the impact of the web. In fact, there have been rumblings for years about how the tenure system no longer serves society's best interest. You've spent many years at an institution that doesn't use the tenure system, and I imagine that that gives you unique insights into the issue. Can you comment?

Yes, I think that tenure has been straying from its original purpose of academic freedom and into sort of a sinecure or something. […]

I don't have a lot of worries about [people having] politically unpopular beliefs, at least in computer science. You know, it's not like [people] have been labeled a Marxist computer scientist and been purged[.]

If somebody did take a dislike to my work, I've noticed that the half-life of administrators is about five years. [It's] going to […] take them a few years to figure out that they don't like [me. W]e're on a rolling contract system, and so if they [wanted] to let my contract run out [and not renew it,] they'd probably be gone before I was. So I'm not worried [about an administrator causing my contract not to be renewed].

The advantages I've seen to [being at an institution without a tenure system:] We do have promotion [at OGI], and I went through promotion to associate professor at the same time that most of the people who graduated when I did were going through tenure decisions. [B]oy, their lives were real crazy, just out of whack[, at tenure time! Promotion] wasn't that big of a deal for me, and at this point if I went to a place with tenure I'd expect [to be given tenure---so I've permanently avoided the hassle of the tenure process].

[T]here's an interesting trend [that] in the last 20 years, the percentage of [tenured] faculty at institutions [that offer tenure] has dropped from about 80% to 50%. What's happened is [that as] faculty have left to retire, they've been replaced by contract employees, part-time employees and so forth. So tenure isn't quite as big a thing [as it used to be].

I was just enjoying reading [the paper] a lot, and having this thought that I couldn't have said it better myself, and then [realizing] with horror that I *had* said it myself … I found out a little later that this paper had been awarded [the] best paper award at the conference where it was presented

[Another] advantage to [not being on a tenure system] is that we don't have a lot of deadwood. [From] talking to people at other institutions, [I know that] tenured faculty can have worse than zero productivity. They could have negative productivity because they [may] like to start committees and start procedures and get in other people's way.

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

I would take a course from somebody else in the department. Here we have all these really smart colleagues that are teaching these great courses. [I] absorb a little bit of [the course material] indirectly, by sending my students [to the courses], but I should go in there [in person] and be refreshing my knowledge.

One of the most useful things I did on sabbatical was [to take] two semesters of graduate [operating systems] from Marv Solomon. [I took the courses] because I hadn't really had an operating systems course since my first year as a graduate student, and [the world has changed

since then]. There's this notion of shared memory and parallel processing and networking that weren't there then, so it was quite valuable to [take the courses].

> One of the nicest things someone ever said to me was … "Boy, you know a lot about systems for a theory person."

Have you encountered situations in your career that were tricky to handle?

Yeah, there was one where I was reading a paper for a database reading group, and was just delighted that someone had figured out really what the important issues were in object databases. I was just enjoying reading [the paper] a lot, and having this thought that I couldn't have said it better myself, and then [realizing] with horror that I *had* said it myself, and realizing that a large chunk of this paper had been plagiarized from one of my papers[. L]ooking at [the paper] more, I found [at] least one other place [in the same paper] that had been taken from another paper. [After I discovered the plagiarism,] I tried to figure out, so what do you do in such a case? [I] wrote [a] snotty letter to the authors, pointing [the plagiarism] out and [saying that the plagiarism] wasn't okay[.] I was going to leave it at that until I found out a little later that this paper had been awarded [the] best paper award at the conference where it was presented. And at that point I thought [that] this [act of plagiarism] should be more widely known, and so I contacted the person who was the program chair for that year to let them know. [You] can't really withdraw the award or anything at that point, although I thought that since about 15% of the paper was mine, maybe I should have gotten 15% of the award. [laughs]

Well, can you list it on your resume[, …]

Yeah, wrote the intro and motivation to an award-winning paper[. T]here was never any public announcement of [the plagiarism], but there was information communicated to departments where the people were, just to say that this was something that needed [have] more attention [paid to it].

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

I'd be more crotchety. I'd be meaner and say no to more things. [I] don't like to make people unhappy, so I have a hard time saying no[. T]hen I end up doing a lot more reviewing and work on program committees and interviews and stuff than I have time for but...

Okay, well that's all my questions for today.

You didn't ask the Paris question ... That was my favorite question.

Is it true that you've been to every museum in Paris? Do you have a favorite?

I haven't quite been to every museum. I've not been to the bread museum yet, but [I] bet I've been to more museums than the average Parisian. [I] guess my favorite [that] I've seen recently

would be the counterfeit museum. [The counterfeit museum doesn't display] counterfeit money so much as counterfeit products. So they have this museum of counterfeit Versace handbags, counterfeit perfume, counterfeit watches. It's just really interesting to go in there and look at what people think is worthwhile counterfeiting.

Can you tell the counterfeits from the originals?

[For] some things, it's painfully obvious and you can't imagine [how] anybody would have bought the counterfeit [while] thinking it was the original. In other cases it's very hard [to] tell them apart. People are very clever; it's very easy now to copy the logo and packaging of the product you're counterfeiting. [T]hey have auto parts there that have been counterfeited, and there it's very [hard], without been a materials tester, to know what the difference is between [a counterfeit auto part] and the real one. You can't see it visually, I don't think.

There [are] some other interesting [museums], like the magic museum and I like the Arts et Métiers museum that has, among other things, Lavoisier's chemical laboratory equipment. Anybody who's interested in museums in France should go to my web page and look at the notes I made while I was on sabbatical at INRIA.

Thank you for listening. Thank you for joining us, Dave.

You're welcome.