

Gio Wiederhold Speaks Out

**on Moving into Academia in Mid-Career,
How to Be an Effective Consultant,
Why You Should Be a Program Manager at a Funding Agency,
the Need for Ontology Algebra and Simulations, and More**

by Marianne Winslett



Gio Wiederhold

<http://www-db.stanford.edu/people/gio.html>

Welcome to the second installment in the *SIGMOD Record's* series of interviews with pillars of the database community. In the last issue we heard from Jeff Ullman, and upcoming issues will include conversations with David DeWitt, Avi Silberschatz, and Hector Garcia-Molina.

This issue's interview with Gio Wiederhold took place in June 2001, a few days before the festivities associated with Gio's retirement from Stanford University. Gio has been a member of the Stanford faculty for many years, active both in computer science and in medical informatics. In the late 1980s, Gio also spent several years as a program manager at DARPA, focusing on middleware and mediators. Gio is an ACM Fellow, an

IEEE Fellow and Golden Core Member, a Fellow of the American College of Medical Informatics, a recipient of the SIGMOD Contributions Award, and a past editor-in-chief of ACM *Transactions on Database Systems*, among many other distinguished positions.

Gio has had a varied and fascinating career, and it is tempting to slip into a series of anecdotes about adventures Gio has had on the job in nations around the world. But instead, I suggest that you ask him about his adventures yourself the next time you see him on the road. Gio loves a lively story and a fun time.

The usual caveats: To let the conversation flow freely and easily, I videotaped the interview and transcribed it later. The errors of transcription, the changes necessary when speech is converted to the written word, and any edits for length are my own. Eventually the videos will appear on the SIGMOD web site, so you can directly hear what each pillar has to say, without my editing and with all the original nuances of emphasis that are eliminated during transcription.

This [interview] is for a new feature in the SIGMOD Record, [and] I thought the occasion of your retirement was a great [opportunity to get a few pearls of wisdom]---

Who else have you interviewed?

Well, no one. [You] will be the inaugural interviewee. Unfortunately it means I'm not very good at [interviewing] yet.

Well, you can experiment with me.

Gio, when you were in your forties, in an established career, with one baby in hand and another on the way, you left an established career as a computer systems manager to become a PhD student. Obviously that wasn't an easy thing to do; what made you do it?

“[Getting my PhD] was also a bit of an experiment, because I didn't have any degree at the time.”

In some ways, [at my job] I was moving from technical things to management. The management problems were always very repetitive. I was always dealing with people, and people haven't improved much over the past millenium. They have problems of not working, not doing the right thing, skipping out on you, getting into useless arguments, so I guess I preferred working with machines.

Getting a PhD was the only way to continue to [work with machines]?

Yes, to move more into research, to both move up [and move away from management.] It was also a bit of an experiment, because I didn't have any degree at the time.

You had an aeronautical engineering degree [from Holland], didn't you?

Yes, but it was roughly at the level of an associate's degree.

I see. So you went directly from your associate's degree to your PhD.

I had some good recommendations from people like [Joshua] Lederberg ... if you get a recommendation from a Nobel laureate, [it helps a lot with PhD program admissions].

You've been a professor in the CS Department at Stanford for over twenty years now. Can you describe your appointment there? [You're] a professor of...

I switched a bit back and forth between full time computer science and part time computer science and [part time] medicine. Partially that was due to limits that Stanford had on research professors, [so] that only 10% of the faculty could be research professors.¹ [Stanford's Computer Science Department] had something like 26 faculty members, and Bruce Buchanan and, I think, Tom Binford were both there as research professors at that time. But of course my PhD is also in medical information science, and I had [at that time] an equal number of publications in medical computing [and in computer science]. [Medical computing] has always been a very motivating application field.

What piece of work of yours are you the most proud of?

[It's] something I did before becoming a faculty member: The ACME time-sharing system that I built for the [Stanford] medical school.

What was special about [the ACME system] at the time that you built it?

It was very well integrated, so that, [for example,] the editor was just a function of the compiler. It was a one-language system; we implemented a subset of the PL/1 language, which turned out to be very teachable. The technology was interesting in itself: it was an incremental compiler, of which there are not very many instances [in existence], but which are a nice balance between a traditional compiler and an interpreter. The way it works is that every line of PL/1, whether it's part of a statement or multiple small statements, gets compiled but then inserted into a list structure. So there is no inter-statement

“If we really want to [meet the needs of decision makers], we want [information systems] to go essentially seamlessly from the past into the future.”

¹ Many universities in the United States have two types of professor positions: tenure-track and research. The major differences are that once granted tenure, tenure-track professors can only be removed from their positions under exceptional circumstances; the salaries of tenure-track professors are paid by their institution; and tenure-track professors have certain obligations with respect to teaching and departmental committee work. Research professors generally must pay their entire salaries out of their grant money; they have contracts with the university that are reviewed and renewed every five years or so; and they typically have lighter or even non-existent obligations with respect to teaching and committee work.

optimization, but each individual statement is fully compiled. And later, if [the user wants] to change a statement, we simply change the list structure---so we can insert new statements, take out existing statements. The only thing [the user] cannot change [in a program is to] make major TYPE changes. [For example, the user] cannot change a numeric value to a string, because there is too much code [that would need recompilation].

[The ACME compiler's flexibility] was very important for the real time aspects of the system. The system did real time data acquisition. [When a problem arose with a program, the ACME system] allowed [the user] to change [the program] while a [medical] experiment was [still] going on, [without] cancelling the experiment (which often involved animals, etc.) [or] losing data.

[When did you build the ACME system?]

I started in late 1965 and it was running by 1967. We bought an outrageous amount of memory for the time. One megabyte of memory. It was so big that I remember that the truck driver, coming from Poughkeepsie, would call me every night to tell me how far he had gotten with that huge memory.

The people who used [ACME]---what kind of a system were they used to? Some of its features were still novel today, but back then it must have been much more...

“My recipe [for consulting is that] you spend a day only listening, then you think about what you heard. The next day you ask questions and listen. And only on the third day do you start giving any advice.”

It was the first computer that most of them had used! And later, in fact, I met medical people who went to new and better systems, and they were frustrated that they would lose data when there was an error in their program, [and have] to start [their experiment] over.

I've heard some people say that all the exciting action in the database research world is in industry now. What do you think about that?

No, I don't think that's true. Maybe the exciting action is no longer in the core areas of databases, but more in the applications. As we get more into applications, there are very many

areas that require research. [For] instance, I've been focusing on an aspect of [information] integration [whose importance] I didn't really realize very early [on]: semantic differences. There has long been work on schema integration. And I learned [from my] consulting [that] people integrate schemas and then the [integrated] databases [still] don't work together because terms mean different things in different databases, [terms] have different scope [in different databases], etc.---so my recent work is on ontology algebra.

What's that?

In order to [integrate] databases or data from different sources, you have to resolve ontological differences. I don't think we want to keep integrating and making these databases bigger and bigger, [as in] a union operation. Typically we want in fact to *articulate* the data, [to just] find out where [the data should be] joined for a particular application. [A simple example of this is] the shoe factory and the shoe store. You don't want to integrate [all of the parts of] the databases [from the shoe factory and the shoe store], but in order for them to cooperate, there has to be agreement on matching in the [relevant] intersection [of the two databases], namely shoe sizes, shoe styles, shoe terms, [those things] that the [shoe store's buyer] understands; but you don't have to integrate the personnel [relations] of the shoe store [database] with the personnel [relations] of the shoe factory [database]. In fact, [integrating] some things [based on] simple term matches would be greatly wrong---like the nail in the shoe store is part of the anatomy of the customer, and in the shoe factory it's something to hold the heel on. So these simplistic matches just aren't good enough.

[Also, when] you get data from different sources, there are differences in granularity. My example is [that] when I work here [at home] with my son, and we need some nails, I tell him to get a nail from the coffee can that's yay long, and has a big head, [while] a carpenter uses terms like "box nails". Carpenters have as many names for nails as Eskimos are supposed to have for ice. And that's efficient. It would be inefficient for me to have to learn all the carpenter's terms. It would be inefficient for the carpenter to talk my terms.

All these kinds of transformations of ontologies can be formalized, and that's where the intersection, projection and similar operations come in. But [these transformations] would be knowledge based operations because they contain matching rules. I think that [that approach to handling ontological differences] provides very flexible scalability, where the [data] sources [can] remain autonomous and efficient.

Do you have a paper on [handling semantic differences through ontological transformations] already?

There have been some early papers on this. Jan Jannink finished with an early thesis on it and Prasenjit Mitra is doing more of the formalization. But to a certain extent, because [the research area] involves semantics, there can be easily ten more years of research. It's not quite traditional databases, but it's certainly necessary in order to make [databases] useful for the kind of decision making support, and for all the other kinds of things, that people say that databases are good for.

“[Being a program manager] is a way to change [the world] to some extent, [and to change the direction in which] things are going ... [Bureaucrats] are not the right people to [act as program managers].”

Did the idea for [an ontology algebra] come from your consulting work?

Yes, because I get called in often as a consultant because I've written about things and then people try to do [whatever I've written about], and then [from consulting for those people, I] find out [that what I wrote was] wrong.

Do you think that all database professors should be out consulting?

Some of the time, if they're capable of doing consulting. Consulting requires very careful listening. My recipe [for consulting is that] you spend a day only listening, then you think about what you heard. The next day you ask questions and listen. And only on the third day do you start giving any advice. There are what I call recipe consultants, [who] know already what the right thing to do is when they walk in [the door]. Sometimes [recipe consultants] can be useful, but that's not what I think is effective consulting, and [with recipe consulting] you [also] learn much less. I've certainly learned a lot through consulting, and I still do.

Gio, you have an impressive record as a prognosticator of what will become important. Mediators is the best known of these [predictions]. What areas do you [currently foresee as] becoming important?

An area that doesn't exist yet, but yet could become potentially just as big as databases, [is] combining [databases with] the results of simulations, [even] simulations as simple as spreadsheets. [At] the moment we say we are using databases for decision making, and actually [databases] only give us a past history. A decision maker also has to project into the future. [Decision makers need] an information system [that combines] database technology and methods for projecting into the future, which would take something from planning technology. Most planning [done by decision makers] today is very data poor, and so it is often not very precise and also very hard to modify, so that plans that are being done are often very inflexible. So I think that if we really want to have information systems [that meet the needs of decision makers], we want to go essentially seamlessly from the past into the future. But the research that's required [to make this happen] requires new insights. There are some very hard technologies---actually, Marianne, you know about that, because [you worked on the topic of] multiple alternative futures.

True, true. [Handling multiple alternative futures is] hard. [But] there's only one past, theoretically.

Unless you have historical revisionism.

I remember you talking about [this same research topic] back in the mid-80s. So when will its time come? [In the case of] mediators, you ended up going to DARPA and pushing to make that happen. Are you going to do that again for simulations?

I don't have the energy to go to DARPA again and push that. But I do see the need [for work in this area]. Certainly if I talk to people that are at the decision making level, they understand [the] problem [that simulations can help to solve]. In my retirement, I won't be completely idle---

“All the internet [comparison] shopping [services are] based on price comparison, but for many things that we buy, price is a secondary issue.”

I doubt you'll be idle at all!

---and I'll continue to push some of these things. But I want other people to do the research, because within my remaining lifetime, I can't see the end of that [research].

That ties into another question. I heard that NSF has 87 program manager openings right now. As a former DARPA program manager, [why do] you think people should be interested in those kinds of opportunities?

DARPA also has plenty of open positions and currently an insufficient number of academics.

If you leave all the [job of giving out the] funding for the future [computer science research] to essentially bureaucrats... A bureaucrat typically gets hurt if [the bureaucrat does] something wrong, and gets very little benefit out of taking risks. So [bureaucrats] are not the right people to [act as program managers]. I know it's very hard to [be a program manager], and especially now when you have many families where both people work. You cannot do it casually. Being a program manager part time is just not effective. But [being a program manager] is a way to change [the world] to some extent, [and to change the direction in which] things are going. [To make that happen,] you have to be willing to go there [to an agency,] and fight [for your program] and take some risks.

You have the advantage, [if you are an academic,] that you can always go back [to academia after being a program manager]. The biggest negative thing I found [about being a program manager] was the time it took [after returning to academia to] restart in research. [Yes], I could get some funding [to restart my research program], but then [I also had] to get the students. Normally you have a continuous pipeline [of students], and you need both the students and the funding [if you are going to conduct research]. [Restarting my research program] took a year more than I had expected.

What do you think is the most exciting trend in technology now, and why?

Well, obviously, the very broad access you have to information sources now. And that, of course, makes the ontology algebra [that we talked about earlier] more relevant, because the [information] sources will be autonomous. [We] will probably have a fairly common representation [for information], in terms of XML, but that still leaves all the semantic issues open.

[A related exciting trend is the] move to on-line publishing, [which opens up] some very hard [problems], [such as] in the [e-commerce] world, [how] to have good metrics for quality. [At] the moment, all the internet [comparison] shopping [services are] based on price comparison, but for many things that we buy, price is a secondary issue. If they are fungible items like books, then [the primary issue is] price. When I buy a computer, or I buy a car, then price is only one of the factors, and quality and long term maintenance and usability are much more important [than when buying a book. If] I want to buy a projector, for me an important fact is how much noise does [the projector] make. And I cannot at the moment use the internet to find out differences in noise levels of projectors. [And of course the noise level] is just one quality to measure. Reducing everything to [a comparison of] price [oversimplifies] our value systems, also.

What changes would you like to see in the tenure system, or the whole university system?

I switched from being in the tenure [track], not quite voluntarily. But I wasn't a good enough teacher; essentially, I didn't get good enough teaching reviews. A non-tenure track position was better] in many ways, [because] I could say no to teaching courses that I knew I couldn't teach well because I wasn't interested in them, or [that] I wasn't in fact interested in teaching the way they were being taught. I remember courses that were advertised as computer architecture courses, which turned out to be assembly language courses, using machines with horrible architectures, so you could never develop any enthusiasm for [the subject matter].

I don't think tenure is as important as it's made out to be, certainly not at a research university, because your own salary is just a fraction of the support you need to run a reasonable research establishment. So I think universities could do without tenure. I don't have the strongest feeling about changing [the tenure system, however].

What words of advice do you have for fledgling or mid-career database researchers and practitioners? You've been on both sides of the fence.

I became a database practitioner for a very pragmatic reason. [When I developed the ACME system,] I built a real-time data acquisition system that collected so much data that people couldn't manage it very well. And so I used what was available in terms of database technology, expanded on it a bit, and built essentially the relevant system [for the problem at hand]. So for the fledgling [researcher or practitioner,] because databases is such a large field, [I recommend having] some contact with some set of applications. In my case, they were medical and military applications. [The medical and military applications] seemed very different. But in both cases, you have to be very responsive to people that have to make rapid decisions under uncertainty. So in that sense, [the medical and military applications] are different from many business situations, in which the decision maker never needs direct access to the computers because [the decision maker has] middle managers that prepare reports and assessments.

“When you are in industry, you shouldn't try to imitate writing academic papers.”

[Whatever application you choose, understanding] it involves a willingness to spend some fraction of your life understanding the application area in depth.

And for practitioners, [I recommend that you] learn enough of the theory and keep up with what's happening in the field, read a bit. [I think that] the reason I could move into academia, even though I hadn't planned to move into academia, was that as a practitioner, once a year I wrote up what I was doing---[starting] from some of my very early projects [investigating] rocket fuel combustion, etc., from the late 50s. They are not the greatest papers; in many ways, they are papers that show just what [I] did, with some quantitative information.

When you are in industry, you shouldn't try to imitate writing academic papers. You should benefit from what you do in industry, where you can often report quantitative data, [and] you've worked on larger scales. Getting those numbers out, and [getting those] measurements out, is very valuable for the academics. Your papers will be read much more than if you write a third rate academic-looking paper.

Do you see that [difference between academic and industrial papers] in the program committees [that] have an industrial track?

Yes, and I've often tried to give [authors] feedback, when I was on a program [committee's] industrial track: you have a nice description, [now] give us some numbers from what you did, rather than the underlying philosophy that made you successful!

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

I've been mainly a user of theory; I can use theory and apply it. I would like to have been stronger on manipulating theory. Luckily, between having good students and good colleagues, I've been able to profit from them [when my research had a theoretical component].