Peter Chen Speaks Out

on Paths to Fame, the Roots of the ER Model in Human Language, the ER Model in Software Engineering, the Need for ER Databases, and More

by Marianne Winslett

Peter Chen

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're in San Diego, site of the 2003 SIGMOD and PODS conferences. I have here with me Peter Chen, who is the Murphy J. Foster Professor of Computer Science at Louisiana State University. Peter is well known as the inventor of the Entity-Relationship model, which is widely used in industry for modeling data. Peter's 1976 TODS paper on the ER model is one of the most frequently cited articles in computer science, and has been selected as one of the most influential papers in computer science by a large survey of college professors. In recognition of his contribution, Peter has received the IEEE Harry Goode Award, the ACM/AAAI Alan Newell Award, the DAMA International Achievement Award, and the Stevens Software Method Innovation Award. He is an IEEE, AAAS, and ACM Fellow. Peter's PhD is from Harvard University. So, Peter, welcome.

Thank you. Glad to be here.

Peter, you're one of the best known of database researchers, yet your career has followed a different path from that of the other people I've interviewed so far. You found a route to fame that does not involve writing zillions of papers! How did you become so successful? And, more generally, what does it take to be a well known and well respected database researcher?

As you mentioned, there are different paths to fame. I think that either you write zillions of papers, or you write a smaller number of papers, but most of them important. So I think the key thing is to solve a real problem, to solve the problem people are worried and concerned about. And I just was lucky enough to solve at least one of those problems.

Do you have any tips for young people who'd like to be able to figure out what one of those big, real problems is?

I think many people have their own visions of the world, their visions of what important problems are. But sometimes because of all the external pressures, they cannot devote their energy to those problems. So I would suggest to many young researchers that after they are established or get tenure, they should devote much more time to their visions, because not only should they develop their visions, but they should follow their gut instincts, their conviction. Do the thing you believe in, and believe in the thing you do.

Do the thing you believe in, and believe in the thing you do.

Why has the ER model been so successful?

I think the reason is that a lot of people don't know how to organize a database, how to organize their data. The ER model gives them a very natural way to organize the information. The concept of an entity and a relationship is one of the most fundamental concepts in our minds. When we look out at the world, what do we see? In this interview room, we see a video camera, we see people, we see chairs, right? You are sitting on a chair and we are facing a camera together. So that is the physical relationship between you, me, a chair, and a video camera. And we also are in the same room together at this particular instant of time. That's a time and space relationship between the two of us. So, entities and relationships are a natural way to organize physical things as well as information.

One of the unique characteristics of the ER model within the world of common database techniques is that it conveys information visually. If a picture is worth a thousand words, should we be using graphical techniques more widely in the database community? And if so, where, and why aren't they already in use?

I think that the use of pictures is very important in many, many areas and that we [computer scientists] should have used them much more. I think that because of a culture difference, because of other things, we have not so far used [pictures] as we would like to have. But I think that as we evolve into the future and computer technology can support graphic things much more easily, we will see much, much more in the way of graphical interfaces and iconic languages.

Entities and relationships are a natural way to organize physical things as well as information.

In terms of database research, what should people be looking at?

Instead of the textual type of database we have today, in the future we should have a graphical type of database, both in the [user] interface and also in the natural structures.

Do you have any instincts on what a natural graphical interface to a database might look like? I guess I haven't seen any organizing principles for that yet.

Instead of typing and using the keyboard, we probably should use your personalized icons as the interface. Let's say you have an icon representing a person, and you have different small icons which represent different individuals. So then those are the interface; but internally, we can also structure data in that way.

Let's talk about the concepts that underlie the simplicity of the ER concepts. What linguistic concepts are behind the ER diagram concepts?

The ER model matches very closely the structure both of English and of many other natural languages. English has nouns and verbs. Nouns usually correspond closely with the entity concept in the ER model, and verbs correspond very well with the relationship concept. In English we also have the concept of adjectives and adverbs. Adjectives correspond to the concept of attributes of an entity in the ER model. Adverbs correspond to attributes of relationships in the ER model. So there is a close correspondence, a one-to-one type of association.

Is there also a relationship between ER modeling and the ancient Egyptian pictographic languages?

Yes, that's the reason I spent a lot of time looking at ancient Egyptian languages in the last few years. Actually, I have probably 20 books on Egyptian hieroglyph languages.

The ER model matches very closely the structure both of English and of many other natural languages.

Wow.

I probably should teach a course on that. It turns out that the Egyptians also developed some kind of pictorial language, similar to Chinese characters, to represent the real world. They have different icons for different objects in the world and different events, and also represent the relationship between different entities. And interestingly enough, there are some concepts very similar between ancient Egyptian languages and the Chinese language. There are several characters, like "sun" and "water", that are exactly the same symbols. So it's very interesting to see that people thousands of miles away still think the same way. In terms of correspondence with the ER model, both languages have not only the principle of modeling the real world, but also the principle of composition. So you can compose two concepts to become another concept.

I've heard that some Chinese dialects are some of the most difficult languages to learn in the world. Is that related in any way to ER modeling?

The Chinese language is probably one of the most difficult to learn, in terms of pronunciation.

How about the written language?

What I mean is that when you see the character, you don't know how to pronounce it. However, in order for people to memorize 50,000 words, there are some simple principles which can help and that match very well with ER modeling---for example, there are principles on how you compose things together and also the concept of type versus instances. Those things are used very often in the information technology field, and it's very natural in the Chinese language.

Is it possible to take additional linguistic concepts and improve our current conceptual modeling techniques further?

It is possible to do that. I have been trying to investigate the concepts available there because I believe that in order for any information modeling methodology to succeed, it needs to be very natural to human thinking. The linguistic concepts have been practiced for many years, so maybe there are some concepts there that could be borrowed.

The relational model has been very successful. Are tables a fairly natural construct for humans? Is that part of the reason for the relational model's success?

Yes. The table is a very natural concept. In many places you see tables, you see matrixes like the spreadsheet.

Did the ancient Egyptians use tables? Have they been found?

Not yet. I haven't found that yet. But what I'm saying is that tables are one of the natural structures, but not the only natural structure. The relational model gives us one particular form of data structure, but we need to worry about other forms, too.

And do the other forms correspond to, say, object oriented models or any other models that we've worked on so far?

The object-oriented model is another form of data structure. However, I think that the ER model actually is on top of both the object-oriented model and the relational model, because the ER model is more conceptually oriented. The object-oriented model and relational model are more implementation oriented. So the choice between those two models is the choice between two different ways of implementation.

What are the major open issues today in the field of conceptual modeling?

There are many, many issues! That's the reason we have the annual conference about conceptual modeling [(http://www.cs.fudan.edu.cn/er2004/)].

One open issue is the most natural way to represent information or data structures. Is it binary or n-ary? Do the relationships have a direction or no direction? Which structures are language dependent, linguistic dependent, or culture dependent? A structure may be very natural to English speaking people, but may not be natural to other kinds of people. Or maybe if we find some living creature from outer space, maybe they will think differently.

I'm sure they would.

ER modeling has been proposed for use in software engineering. Has it found users there, or is that still in the research stage?

A lot of people have used the ER model in software engineering. It turns out that when people develop an application, the two most widely used diagram techniques are ER diagrams and data flow diagrams. So ER modeling has become a standard technique in many large scale systems.

What do you think of the newer modeling languages like UML?

UML actually is very good in many aspects, but has its own limitations. On the positive side, it helps to validate the ER concepts and to bring the attention of OO programmers to the importance of conceptual modeling. My personal opinion is that UML is a language and graphical convention. It helps to establish the ER concept yet doesn't replace the ER concept. It's very important to recognize that, because I can view UML just like any conventional high-level language, like ALGOL, PL/1, FORTRAN, and other different kinds of languages. In each of

these languages you have the concept of control structures, data structures, assignments, and whatever. Those things are always there in programming languages. Similarly, UML is a language that implements key ER concepts with an OO flavor, but UML is not and will not be the only language implementing major ER concepts. In short, the ER concept is the basic fundamental principle for conceptual modeling. That will be always with us, has been with us since thousands of years ago and will be with us for many years to come.

How can the ER approach help in XML modeling?

XML is basically a tree-structured type of language. I think the XML community understands the need for richer structures, so they are developing modules to enhance that. I'm associated with several working groups in the XML community, serving as an invited expert in the XML schema working group and the Xlink working group of W3C. For example, in the XLink group, they are trying to develop the hyperlink into a more sophisticated linkage structure. We can view the extension of a hyperlink to a higher level structure as trying to implement the concept of relationship from a low level relationship to a high level relationship; so the evolution is similar to moving from a physical pointer to a conceptual pointer in the evolution of operating systems. Furthermore, it also evolves from a link between two resources to a link between more than two resources.

So it could be similar to an M to N hyperlink, and perhaps have its own attributes?

That's correct.

What has been the main impact of the ER conferences over the last 25 years?

The ER concept is the basic fundamental principle for conceptual modeling. [It] has been with us since thousands of years ago and will be with us for many years to come.

Number one, it provides a nice place for people to exchange ideas and to develop new ideas. It also established a new field of study. It is a way to have different kinds of people come together. The attendees are not just people from the database field; people come from many different fields, such as management information systems, software engineering, AI, and information retrieval. Sometimes people present ideas on using the ER model in music design or organizational design; so the conference is a way to expand the discipline. I think the conferences have proved to be very useful in helping the field and establishing conceptual modeling as a respectable field of study.

How can ER modeling help with interoperability?

We have built a lot of systems, and many of the systems don't talk to each other. There are many reasons for that, but one key thing is that the data structures are not compatible with each other. What we have today is similar to having many islands of isolated information, and we really need to build links between those islands. In order to build those links, we need to discover the underlying entity relationship structure. Sometimes the structure is hidden, sometimes implicit; so, in many cases, data mining techniques are needed to try to discover the underlying relationships between entities.

How can structural modeling be extended to functional modeling?

First you need to have structure, and then you can add the functions, the manipulation part, to the structure. Today in the industry, people are doing these two things separately. They are doing data modeling, information modeling, in one group, with another group doing functional modeling. Their models may not be comparable, and the people are different. What I envision in the future is both kinds of modeling being done by the same group of people, using the same techniques, so that then you have one model which covers both the data aspect and also the functional aspect.

Would the people doing the structural aspect be using, for example, an ER approach?

That's what people are doing right now. They're using ER or ER-like structural techniques, then a different kind of technique for the functional aspects.

And what kind of techniques are they using on the functional side?

They are using dataflow diagrams, state transition diagrams, or whatever. I suggest that in the future, we can take the ER concept and add something onto it so that it becomes a unified modeling technique. Then you will have one technique that can be applied to both structural and functional modeling.

Okay. So that would be the ERF (Entity Relationship Functional) model.

Right. Good. You already have the name.

Okay. You just have to write the paper.

Right.

Why have ER database management systems never become popular?

I remember using an ER model database management system at Xerox PARC in the 1980s. Given the wide use of the ER model, why have ER database systems never become popular?

We have to recognize that there are technical, political, and economic issues involved. The main reason is that the timing wasn't right. People had the right technical idea, but the timing wasn't right for it. In terms of economic issues, the problem is that there's a huge investment in certain technology, established technology. Twenty-five years ago, there was a large investment in IMS hierarchical database technology. You probably heard how Dr. Ted Codd had troubles in convincing the IBM management to implement relational technology. He was very frustrated with the IBM management, and had some bitter words about that. Now the relational DBMS technology has become the mainstream technology, and industry has invested much, much more in it than 25 years ago. So with the established forces there, and also the inertia there, it would be a little bit difficult to introduce a new thing on top of that.

So that was also a part of the problem for object-oriented DBMSes. The academic world talked a lot about OODBMSes, but the commercial world had very little response because of the heavy investment in relational technology. However, I see that the timing is changing; it's becoming right for the ER type of DBMS. I think in the next ten years, you will see a lot of research on that, and possibly implementations and even commercial systems.

What's the difference between an ER DBMS and an object-relational database system?

The object-relational system is still concerned with objects. Now, what is an object? I think the "object" is basically, if you look at the word itself, a very high level concept; but the way it's being implemented today, objects are basically the wrappings of the functions together with the data they manipulate. Because the concept was used in such an extreme way, it makes some of the functions not very natural. Let me give you an example. How can you implement 2+2 in an object-oriented system? You send a message that says "+2."

It does seem a bit awkward, doesn't it?

Right. That's awkward. Certain things cannot be naturally expressed in an object-oriented way. So that is causing some of the problems. My point 25 years ago, and my point today, is that you must do things in a natural way. Anything unnatural will be very difficult to implement and very difficult to get accepted in the commercial world and by the public.

Certain things cannot be expressed naturally in an object-oriented way. My point 25 years ago, and my point today, is that you must do things in a natural way.

So the object-relational databases are still in the unnatural space?

Right, but my comments on the unnatural space are more towards pure OO DBMSes than certain object-relational DBMSes. Some of the object-relational DBMSes are more of a marketing ploy than real attempts to implement OO DBMSes. If an object-relational database can avoid implementing those unnatural/awkward features, I don't have any negative opinions on it.

For the ER databases that you see in the future, what applications do you think will be driving their acceptance?

The applications driving their acceptance would be the high-level matching of concepts. So for example, suppose we want to find terrorists. Who are the potential terrorists? You may know whether somebody purchased fertilizer, and whether money is being channeled someplace else. These are complicated relationships. The system can come up with questions to supply you with potential answers to your high-level query ("Who are the terrorists?"). That kind of processing can be done naturally in an ER type system.

Why would it be easier there than, say, in a pure relational system?

In the pure relational system, the linkages are not explicit. Also, the data are scattered around, and you don't know what matches with what. Sometimes you will match with the wrong thing. You could have the name of person and the name of a ship, and you match them together, building a wrong type of linkage. In an ER type of system, you will avoid this kind of problem; you would not build an incorrect linkage. We have tons of information, and we don't have time to worry about incorrect relationships. We cannot spend extra time to collect incorrect linkages and irrelevant types of data.

More generally, you are interested in techniques that can be used to identify terrorists. What do you think of the privacy concerns that have been raised about the US government's total information awareness program?

There is a trade off, security versus privacy, and we will find out it's a very strange world we are in. Let me give you an example. I live in Louisiana, and we have several problems which have attracted national attention such as serial killers, serial snipers, and everything---you name it.

Wow. Let's have SIGMOD in New Orleans.

Right, SIGMOD in New Orleans, we'll do that.

In the last 24 months a lot of major stories happened in Baton Rouge, Louisiana. In the past, CNN or USA Today would say "Baton Rouge, Louisiana." Now they drop the name of the state, "Louisiana", because Baton Rouge has become a famous city. We have this serial killer, and he can be found quickly if we have the facility to monitor a lot of data, to integrate a lot of data together. But we don't, so a lot of energy and police time were wasted and many people died in the process. They finally caught a suspect a few weeks ago, using DNA. I had suggested that we needed a DNA databank. However, that's a very politically sensitive issue, and is directly related to what you just said: privacy versus security. Fortunately, we don't need to use any technology that causes this type of controversy because we have not yet utilized the legally available technologies today to the full extent. For example, many of the 9/11 terrorists have previous records, for example, have traffic violations; one person was stopped by Maryland police two days before the hijacking. They stopped the person, but they didn't know that he was on the CIA watch list, so they let the person go. Then another famous 9/11 terrorist, Atta, was stopped in Florida. There was a warrant out for him, but the police didn't know about that. There is no privacy issue there; all the information is publicly available. The problem is that the data do not link with each other, so the relationship was not built there. Maybe total awareness is too much, but to be more aware of what's going on by using existing legally obtained data and linking all things together, making inferences using the technology available and following the law--- we should be doing that, and I don't think that will cause any privacy concerns.

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

> I have been very lucky, very fortunate, and very blessed with opportunity. ... I just happened to be in the right place, right time, with the right idea.

I would develop a comprehensive theory of the entity relationship model, making it more sophisticated and applicable to many more domains. The data, the structure, would have multiple levels and have different algebraic operations to manipulate it. The model would be well developed along the lines we discussed earlier, with functionality and data modeling both integrated. We would have not only the mathematical part but also the graphical interface, which will be useful to both schema designers and end users.

I would also like to try to spread the concept of the ER model to other domains, not just in information technology but also other places, such as human relationships. How do you manage human relationships? In the business world, they talk about customer relationship management. Once people look carefully on the relationship side and understand it better, they would manage better, they would appreciate each other better, and they would do much, much better than before.

When you try to develop a theory, you want to make the theory applicable to many different domains. When you look at different domains, you also get feedback that will be useful in the modification of your theory, its structure, and other things. I was just sitting in the WebDB workshop and I heard Joe Hellerstein talking about discovering structure in different fields. That's the approach I would like to take, and I think many people would like to take too: to look at things outside our usual world. We are too concentrated on data processing right now. If we look at other areas, we can discover different kinds of structure; maybe those structures would be very useful. We might find a new functional aspect of that type of domain that we can use in the conceptual model. So those are the things I would like to do.

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

Not a thing! I wouldn't like to change anything because I have been very lucky, very fortunate, and very blessed with opportunities. Sometimes I think I have been too lucky! I ran into Phil Bernstein at the ER 2000 conference in Salt Lake City, and he said to me that the ER concept is a once-in-a-lifetime opportunity, and I fully agree with him. I just happened to be in the right place, right time, with the right idea. It may not ever happen again in the future, but it happened at that particular instant in time. For me, it's very lucky, very fortunate, to have had that opportunity. So I don't think I want to change anything. I really appreciate all the good things that happened to me, and I'm very glad to have the opportunity to contribute some of my ideas to the advance of technology.

Thank you very much.

Thank you.

I have good news for our readers: we have finished the first video version of an interview in this series. By the time you read this column, I expect that the video version of the interview with Hector Garcia-Molina will be available for download from the SIGMOD web site, with more interviews to follow in coming months. The video and print versions of interviews are really quite different, so I urge you to take a look at the videos when you have a quiet moment.
