Reminiscences on Influential Papers

Richard Snodgrass, editor

This column celebrates the process of scientific inquiry by examining, in an anecdotal fashion, how ideas spread and evolve. I've asked a few well-known and respected people in the database community to identify a single paper that had a major influence on their research, and to describe what they liked about that paper and the impact it had on them. Some of the papers listed here have actually changed the career course of their readers. The clustering is also interesting: most of the papers are about twenty years old, but two were published within the last five years.

Laura Haas, IBM Almaden Research Center, `laura@almaden.ibm.com`

[P. Selinger, M. Astrahan, D. Chamberlin, R. Lorie and T. Price, "Access Path Selection in a Relational Database Management System," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 23–34, Boston, 1979]

Why has this paper been so important to me? First, because it is the paper that reconciled me to working in databases. I hated my database course in graduate school, which left me convinced that there was nothing more to the field than boring data modeling issues. Instead, I studied distributed algorithms, and eventually ended up doing a thesis on distributed deadlock detection. This clearly had some relevance to database systems, and I was fortunate enough to land a job at IBM on the basis of that connection—which I almost turned down, because it meant working on a database project. However, I accepted the position, and soon learned that there was more to databases than data models! Among the many papers I read in that first year or so at IBM was this one, and it was the first that I remember reading in the field that made me think there actually were some interesting problems, and perhaps even issues which I could someday help to solve. Little did I know then that I would spend a big chunk of my career on query processing issues in general, and eventually end up working in the area of optimization. Of course, now that I am actually working on optimization, this paper is the Bible—not in the sense of something I look at every day, but in the sense of a set of guiding principles and fundamental rules that shape my approach to my day to day work and research.

Alberto Mendelzon, Computer Science Department, University of Toronto, `mendel@cs.toronto.edu`

[A. V. Aho, C. Beeri, and J. D. Ullman, "The Theory of Joins in Relational Databases," *ACM Transactions on Database Systems*, 4(3):297–314, September 1979]

In the last issue of the *Record*, Jeff Ullman remembered the database course taught by Catriel Beeri at Princeton in 1977 or so. This paper (submitted to TODS in March of 1978) was one of the first products of the ferment generated by Catriel's course. The question addressed was: when does a set of functional dependencies guarantee that any relation that satisfies it can be decomposed without loss of information? The special case of decomposing one relation into two had been solved years before by Delobel and Casey and by Rissanen, and incorrect generalizations of this result, made by well-known researchers, were circulating in manuscript.

As a graduate student, reading an early version of what became known as the ABU paper, I was struck by several facts: that database theory was subtle enough that well-known researchers could make mistakes; that the mysterious phenomenon of "the connection trap" bandied about at the time could be cleanly formalized and analyzed; and that humour was allowed, so that Theorem 2 was called "the Mickey Mouse Theorem" for reasons that are obvious if you look at the accompanying figure (regrettably this name did not make it to the published version).

The simple and elegant algorithm for testing losslessness in this paper, which Jeff and Al Aho used to describe as "chasing down dependencies," served as a starting point for Shuky Sagiv, Dave Maier, and me, all graduate students at the time, to start what became a whole body of work on the *chase* method, still an important theoretical tool today. In fact, at about the same time as this issue of the Record ships, a paper that applies the chase to the highly *au courant* topic of information integration is being presented at the ICDT'99 conference in Jerusalem, which is co-chaired by none other than Catriel Beeri.

Meral Özsoyoğlu, Department of Computer Engineering and Science, Case Western Reserve University, `ozsoy@alpha.CES.CWRU.Edu`

[P. A. Bernstein and D-M. W. Chiu, "Using Semi-joins to Solve Relational Queries," Technical Report No. CCA-79-01, Computer Corporation of America, 1979. (Also in *JACM* 28(1):25–40, 1981)]

This paper had a major impact on my research. When I first read Bernstein and Chiu's paper as a technical report in 1979, I was a graduate student at the University of Alberta, and my research advisor had just moved to Chicago. At the time, I was trying to find a thesis topic in query optimization, and had read several papers in query processing and distributed databases. I had noticed that some queries are inherently more costly to process than some others, but didn't quite figure out how to formalize. Unlike other papers that were based on heuristics, Bernstein and Chiu used a very novel approach: they classified queries as tree and cyclic queries, introduced the semijoin operator, and showed that while tree queries can always be answered by semijoins, cyclic queries may not be. They also gave an algorithm to determine whether a query is a tree query or not. I was surprised to see that this algorithm was not applicable to some example queries that I had identified earlier as "typical" while trying to come up with a query optimization scheme. This motivated me to start working on a generalized tree query membership algorithm and I ended up doing my thesis on distibuted query optimization using semi-joins. Our tree query membership algorithm (co-authored by my PhD advisor C. Yu) was published in the same year in IEEE COMPSAC'79 conference. (Bernstein and Chiu's algorithm was limited to at most one join attribute between any two relations, i.e. semijoins involving single domains). This was my first paper as a graduate student and was a starting point for my research. This tree query membership algorithm was later called the "GYO Reduction" in the literature.

Bernstein and Chiu's novel approach to query processing and semijoin reductions influenced my research, and the research of many others in the area of query optimization. Both semijoins and the classification of tree and cyclic queries are still used today in query optimization almost 20 years after they have been introduced by Bernstein and Chiu.

Jan Paredaens, Department of Mathematics and Computer Science, University of Antwerp, pareda@uia.ua.ac.be

[A. K. Chandra and D. Harel, "Computable Queries for Relational Data Bases," *Journal of Computer and System Sciences* 21:156–178, 1980]

In this paper an abstract characterization of the class of queries which are computable is defined. Its main result is that the completeness of a database programming language can be thought of as consisting of the relational algebra augmented with the power of iteration. This is the basic paper that discussed for the first time the concept of computable queries and made clear the difference between a query language and a general purpose programming language. It also introduced what is later called the genericity, a property that every pure query language has to fulfill. The paper inspired a lot of authors later on in defining more database query languages, in searching properties of computable query languages and in defining the genericity in the context of several types of database applications.

Krithi Ramamritham, Department of Computer Science, University of Massachusetts, on leave at the Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, krithi@cse.iitb.ernet.in

[C. T. Davies, Jr., "Data Processing Spheres of Control," *IBM Systems Journal* 17(2):179–199, 1978]

Davies (in collaboration with L. J. Bjork) introduced a single abstract control structure, namely "Spheres of Control" to achieve flexible semantics for almost every aspect of transaction execution: process (read: transaction) atomicity, commitment, dependencies between transactions, concurrency control, consistency, and recovery. In simple terms, a sphere of control can be viewed as a boundary around the effects of an arbitrary set of operations that can be unilaterally revoked or committed. Spheres can be nested, sequenced, or parallelized. Reading this paper today, anyone working in the area of advanced concurrency control and transaction processing is bound to ask "so, what else is new"? What is "new" is that the work reported in this paper was done in the early-mid 70's! Arguably, everything that has since been "proposed"—for utilizing application and data semantics for better concurrency control and recovery—is a reinvention of ideas introduced here, with more waiting to be reinvented. Unfortunately since many of the terms used in the paper are archaic and predate ACID, it does take a certain amount of effort to translate, and appreciate the concepts "hidden" in the paper, in terms of what is well understood today.

So, this is a paper I wish everyone—especially, those who have done work on advanced transaction models—had read before embarking on their work on transactions. As it turned out, I myself came across Spheres only towards the end of our work on the ACTA. It was gratifying to note that we had not fallen prey to the original temptation of inventing yet another transaction model, but had instead developed a formal framework using which one could analyze and synthesize advanced transaction models. Since then, I have been influenced not only by the perspective offered by the concepts underlying Spheres but also by the fact that in developing these concepts, Davies was inspired by how human organizations perform their activities and share resources.

Nick Roussopoulos, Department of Computer Science, University of Maryland, nick@cs.umd.edu

[J. Gray, A. Bosworth, A. Layman and H. Pirahesh, "Data Cube: A Relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals," in *Proceedings of the IEEE International Conference on Data Engineering*, pp. 152–159, New Orleans, February, 1996]

I first heard about a hot west coast operator called "Data Cube" by Gray & Co in November 1995 at the CIKM'95 workshop in Baltimore. I immediately e-mailed Jim requesting the paper and got back the URL to his collection of papers at his new then job at Microsoft. I downloaded the paper and started reading the paper but, to my disappointment, the figures which, in this particular case are worth more than just a thousand words, had a big cross with some Microsoft mumbling underneath. I guess Jim was still in the process of mastering Microsoft software! Never the less, I was able to get the basic ideas before the New Orleans conference where I got to see those figures.

The Data Cube paper is to the area of OLAP and data warehousing the equivalent of what the 1970 Ted Codd's paper was for the relational databases. It formalized the concepts of multidimensional aggregate views and the hierarchies within them. It also set the first ideas on the complexity of the incremental algorithms for maintaining various aggregate functions. This paper tremendously influenced my research on the cubetree storage organization and its bulk update. In 1995, I was working on materialized views with aggregate and other functional abstractions in them. There was no better timing for this. Thanks Jim, Adam, Andrew and Hamid.

Jennifer Widom, Department of Computer Science, Stanford University, widom@DB.Stanford.EDU

[Patricia G. Selinger, Morton M. Astrahan, Donald D. Chamberlin, Raymond A. Lorie and Thomas G. Price, "Access Path Selection in a Relational Database Management System," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 20–34, Boston, 1979]

I'm lucky to be one of the earlier entries in this "influential papers" series because I suspect this particular paper will come up time and again [Ed: I actually received this contribution before Laura's, above. How prescient!]. I believe this paper has influenced me in a different way from most other people—for me it's largely been pedagogical. When I think about the papers that have most influenced me from a research perspective about a half dozen come to mind. When I think about those that have most influenced me from a teaching perspective, this is the one.

My reasons for loving this paper are all over the map: (1) It's 20 years later and we're still using it as a coding spec—we're still building "Selinger-style" query optimizers. In Computer Science, especially in systems, that level of endurance is amazing. The paper is worth studying for that reason alone. (2) As a non-optimizer expert, this well-written paper eased me into the topic and convinced me that query optimization is an interesting and relatively clean area, with lots of fun nooks and crannies to explore. What better vehicle for teaching students about systems building and research? (3) This paper, the papers it led me to read, and the people it led me to talk to, convinced me that query optimization should be included in a core advanced database curriculum to much greater depth than it has been covered in the past. The query optimizer is the heart of the DBMS, and Selinger et al. led me to believe that from an educational perspective we should all understand the intricacies that lie there.

Philip Yu, IBM IAC, T. J. Watson Research Center, psyu@us.ibm.com

[R. Agrawal, T. Imielinski and A. Swami, "Mining Association Rules between Sets of Items in Large Databases," in *Proceedings of the ACM International Conference on Management of Data*, pp. 207–216, May 1993, Washington, DC.]

This paper provides a mathematical formulation of the association rule mining problem, where the association rule problem tries to identify the set of items often appeared together in a transaction. It decomposes the problem into two subproblems. The first one is on the generation of large item sets based on support and the second one is on deriving the association rules from the large item sets based on confidence. This pioneer work has provided an elegant problem formulation that transforms an abstract problem into an algorithmic problem. It opens up a new area for future research. In additional to potential research opportunities for faster mining algorithms and model extensions, I was most intrigued by the issues on the statistical significance of the rules by considering alternative measures other than support and confidence such as collective strength which is a correlation type measure, and also on the on-line interactive generation of the rules to give users more control on what rules to generate and how to specify the parameters.