Reminiscences on Influential Papers

Kenneth A. Ross, editor

This issue's reminiscences include a paper that the contributor didn't (initially) like, and a paper that took seven years to be published. Sometimes it takes a while for a new idea to sink in.

I continue to invite unsolicited contributions. See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Stefano Ceri, Politecnico di Milano, `ceri@elet.polimi.it`.

[V. Y. Lum et al. 1978 New Orleans Data Base Design Workshop Report. IBM Technical Report RJ2554, and Proceedings of VLDB 1979, pp. 328–339.]

The two papers report the results of a workshop attended by 35 scientists, focused on the process of designing database applications. The workshop identified four separate areas of the design process, denoted respectively as "Corporate Requirement Analysis" (Area I), "Information Analysis and Definition" (Area II), "Implementation Design" (Area III), and "Physical Database Design" (Area IV). The main merits of this workshop were the precise identification of the four problem areas mentioned above, and then, within each of them, the further identification of subproblems. In particular, areas II and III were clearly separated by focusing Area II on DBMS-independent issues and Area III on the development of DBMS-processable data schemes. In my opinion, this recognition helped in the establishment of Area II as an autonomous research field, then named "conceptual modeling", characterized by subproblems such as model definition, view modeling, and view integration.

VLDB in Rio De Janeiro was the first international database conference that I attended, and this paper had a great influence on my scientific work. Since then, I learned to appreciate the difference between "core data technology" and the need of a parallel science for putting core technology into practice, by providing models, methods, tools and infrastructures for the deployment of applications which could use such core technology. The need for giving to such "parallel" science a wider recognition is still felt by database researchers these days, e.g., with the VLDB Endowment's "broadening strategy" and the new emphasis on "Information Systems Infrastructures". And I feel that we still need to understand and master the process of Web application design, in a similar way as twenty years ago we needed to cope with the database design process. Back in the eighties, I was involved with several Italian colleagues in the "Dataid" project (1979-1984) on the development of a computer-aided methodology for database design, which was internationally recognized as one of the most relevant projects in this field, and eventually lead to the writing of the book "Conceptual Database Design" (Addison-Wesley, 1992), with Carlo Batini and Sham Navathe, which in essence reflects the subdivision into the four areas mentioned above, and focuses on areas II and III.

Luis Gravano, Columbia University and Google Inc., `gravano@cs.columbia.edu`.

[Ronald Fagin. Combining Fuzzy Information from Multiple Systems. PODS 1996, pp. 216–226.]

This paper is on the processing of queries over collections of objects that have "multimedia" attributes such as a piece of text or an image. Queries over these collections are typically a specification of target values for each attribute. Users rarely expect in return to their queries those objects that match the target values *exactly*. Instead, a more natural answer for a query in this context is a sorted list with a few objects that *best match* the query. Query processing is complicated because each of these attributes is usually handled by a separate "subsystem" with a limited interface (e.g., a text search engine or an image retrieval system). Ron cast this complex problem in a realistic framework and developed an elegant algorithm to identify the best matches for a query efficiently. His algorithm is simple and intuitive, and Ron proved it asymptotically optimal for an important family of matching functions under reasonable assumptions.

Ron forwarded a preliminary version of his paper to Surajit Chaudhuri and me in the summer of 1995. (At the time, I was a summer intern at Hewlett-Packard Laboratories in Palo Alto.) I was just delighted when I read it. Ron's paper was focusing on a main-stream issue for the database community (i.e., efficiency of query processing) but for a much less main-stream problem for the database community at the time (i.e., computing ranked query results). As a graduate student at Stanford, I was starting to address not-so-traditional database problems from a database perspective. Specifically, I was mainly working on query processing over distributed text databases on that novel thing called the World-Wide Web. Ron's paper was a beautiful example of how to reconcile these two worlds. It was immensely influential on the research that I did subsequently, and still is.

Per-Ake Larson, Microsoft Research, palarson@microsoft.com.

[Witold Litwin. Linear Hashing: A New Tool for File and Table Addressing. VLDB 1980. pp. 212–223]

I first learned about hashing from Knuth, Vol 3 — a wonderful book that I read cover to cover sometime in the mid-seventies. At the time, it was assumed that hash tables and hash files had to be of fixed size and to change the size you had to rehash everything. I became intrigued by this issue and came up with a scheme called dynamic hashing that solved this problem by splitting overflowing buckets in the same way as page splits in B-trees. Around the same time, the paper on extendible hashing by Fagin, Nievergelt, Pippenger and Strong was published. Both schemes split pages to completely avoid overflows but then required an index structure to keep track of which pages had been split. Not a very elegant solution - I felt that using an index structure violated the spirit of hashing. Was there some way we could get rid of it?

Witold Litwin sent me an early version of his paper on linear hashing in 1979. I must admit that I though that linear hashing was a bad idea when I first read the paper. Yes, it did away with the index structure but splitting buckets in a fixed sequence without paying any attention to where overflows actually occurred couldn't be a good idea. Or could it? This question was the basis for my research over the next several years, resulting in a sequence of papers — too many perhaps — with various improvements on linear hashing. And, I was wrong — it was a good idea.

Leonid Libkin, University of Toronto, libkin@cs.toronto.edu.

[Jan Paredaens, Jan Van den Bussche, and Dirk Van Gucht. First-order Queries on Finite Structures over the Reals. In Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS'95), pp. 79–87. (Full version in SIAM J. Comput. 27 (1998), pp. 1747–1763.)]

How often does a single paper make you forget your current project, and start a new one, from scratch, with very little background in the area? That's exactly what happened to me in June 1995, when Limsoon Wong visited Bell Labs. We were looking for interesting problems to work on, and Limsoon suggested constraint databases. My first reaction was negative; I was familiar with a very nice definition of the Kanellakis-Kuper-Revesz paper, and some fascinating open problems in the area, but had a feeling that those problems were beyond reach, and thus not too appealing – what if you disappear for 3 years, and then come back and say "sorry, but it was too hard for me"? Having listened to those objections, Limsoon replied that the situation was not as hopeless as I thought, and gave me a copy of the paper by Paredaens, Van den Bussche and Van Gucht (which was to appear in LICS one month later).

If any subject is to attract attention, it must have an appealing definition; but to develop into a theory, it must have its own methods and interesting results. The Kanellakis-Kuper-Revesz paper introduced constraint databases, but it was not until the Paredaens-Van den Bussche-Van Gucht paper that the subject really took off the ground. The paper developed the technique of collapse results, which turned out to be the key for developing the theory of constraint databases. In essence, collapse results tell you that under some circumstances, constraints used in a query are irrelevant. These results let one analyze – and ultimately design – languages for constraint databases.

The paper is short; I read it in an hour and was immediately convinced that constraint databases is a really cool area to work in – and I still think so. However, there was one victim of my sudden switch – I was writing a paper on view maintenance and query optimization at that time; today, almost 6 years later, that paper remains unfinished ....

Tova Milo, Tel Aviv University, `milo@post.tau.ac.il`.

[Serge Abiteboul and Catriel Beeri. On the power of languages for the manipulation of complex values. VLDB Journal 4(4), 1995: pp. 727–794.]

I first ran into this paper while searching for a subject for my Ph.D. thesis. The fact that I spent the next three years digging into algebras and calculi for object oriented databases clearly shows how much I liked what I've read. This was the first clean formalization of the algebra and calculus for complex values (which later served as the foundation for the object oriented model). A graceful extension of the relational languages to this context, preserving central notions such as safety and domain independence while at the same time incorporating the flexibility required by the more complex nature of the data.

But besides intellectual influence, this paper also taught me as a young Ph.D student an important lesson regarding the hard life of publishing. While the INRIA technical report (written in 1988) is perhaps the most cited paper among those dealing with the foundations of the object oriented model, the paper itself was first rejected from a conference (with the only negative comments being a few spelling mistakes), then submitted to one journal (with its refereeing being postponed for several years with editors coming and going till the authors gave up), and only finally, after seven years(!), accepted to the VLDB journal ....
