Reminiscences on Influential Papers

Kenneth A. Ross, editor

See `http://www.acm.org/sigmod/record/author.html` for submission guidelines.

Kevin Chen-Chuan Chang, University of Illinois at Urbana-Champaign, `kcchang@cs.uiuc.edu`

[Philip A. Bernstein, Alon Y. Halevy, Rachel Pottinger: A Vision of Management of Complex Models. SIGMOD Record 29(4): 55–63 (2000).]

Opening a series of concrete works to follow, this vision paper identifies, motivates, and abstracts the problem of model management. It proposes to support "models" and their "mapping" as first-class constructs, with high-level algebraic operations to manipulate. In the winter of 2000, I was a starting faculty at UIUC, and this paper inspired me immensely at the time when I had to create a research agenda of my own. I have always been interested in information integration, on various topics like query translation and data mapping. The area was exciting to me, as it was full of "real-world" problems. However, it was also not hard to see that these problems seemed inherently messy and their solutions inherently heuristic. Probably because many problems remained unsolved, most research works were only able to address separate topics, without a clear context of an overall application.

This paper changed my view in two ways: First, it shows the relevance of studying related integration problems in the context of an overall system (in this case, a model-management system). Second, it shows the promise of deriving principled mechanisms even for messy problems: By raising the level of abstraction to generally capture the notion of models and their operations, this paper motivates well that a hard problem can be treated with a clean and elegant framework. While both of these seem natural, they were at least unusual for information integration, where problems were often thought to be too hard to seek principles and too early to situate in a system context. Although the paper was a vision paper to be concretely realized, it influenced me in attempting to pursue a system-driven research agenda and striving to seek principled solutions.

Jayant R. Haritsa, Indian Institute of Science, Bangalore, `haritsa@dsl.serc.iisc.ernet.in`

[H. T. Kung and John T. Robinson: On Optimistic Methods for Concurrency Control. ACM TODS, Vol. 6, No. 2, June 1981.]

I first read this paper as a raw graduate student, and was captivated by its refreshingly different espousal of a "Don't worry, be happy" approach towards transaction concurrency control, well before this philosophy entered popular parlance. In fact, it went diametrically against the conventional wisdom at the time that locking, with its "nattering nabobs of negativism" world-view (to borrow a phrase from William Safire), was the concurrency control protocol of choice for database systems. Specifically, the paper proposed a new protocol wherein a transaction's execution was divided into three phases: an "optimistic" (i.e. unrestricted) read phase with private updates, a validation phase to assess the transaction's conflicts, and, finally, a write phase where successful validators transferred their private writes to the public database.

Optimistic concurrency control (OCC) was originally mooted in the context of query-dominant systems, where conflicts are relatively rare. For this environment, the intuition was that OCC's RISC-like approach optimized the average case, as compared to the CISC-like approach of locking, where even read-only transactions suffered from lock maintenance overheads. Subsequently, it was realized that the optimistic strategy had another potent advantage — by lazily delaying conflict resolution to the very end, that is, the validation phase, it provided great flexibility in dealing with transaction data conflicts, as compared to locking, where decisions had to perforce be made at conflict occurrence time. This flexibility could be gainfully used to support a global perspective of the conflict situation, rather than a localized and incremental view, even to the extent of dynamically deciding the transaction commit order. As a case in point, I was able to use this power of OCC in my own doctoral thesis on real-time databases, to ensure timely scheduling of transactions with firm completion deadlines.

This paper is a classic example of how the best research is contingent on thinking against the grain, and therefore has a timeless inspirational value over and beyond its outstanding technical contributions.

Sergey Melnik, Microsoft Research (USA), `melnik@microsoft.com`

[François Bancilhon and Nicolas Spyratos: Update Semantics of Relational Views. TODS 6(4): 557–575, 1981.]

Raising the level of abstraction is often the key to finding elegant solutions for hard and elusive problems. In their paper, Bancilhon and Spyratos provide an exemplary study of the view update problem put forth by Dayal and Bernstein in 1978. To get a handle on the "information not visible within the view," the authors treat views as functions over database states and express the view update problem in terms of view complements. Despite the word "relational" in the title, the developed formalization is agnostic about schema and view definition languages, i.e., applies to any contemporary data model, such as object-relational or XML.

The paper inspired me in a profound way. It prompted me to look into other established database problems, such as view integration, query composition, and view selection, and to think of how those can be characterized and generalized in a concise, language-independent fashion. It helped me deepen my perception of metadata management as a research area that combines formal approaches with engineering practice.

Guido Moerkotte, Univeristy of Mannheim, `moerkotte@informatik.uni-mannheim.de`

[Gérard Huet: Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems. JACM 27(4), 797–821, 1980.]

When I was a student I did not like to go to lectures. Therefore, I had plenty of time to read many papers in different areas of computer science. Among all the papers I read, one paper was really outstanding: the paper by Gérard Huet.

This paper exhibits plenty of attributes a good paper should have. Let me list some of them. Firstly, it is free of buzzwords. Secondly, it does not mention challenging new applications. Thirdly, the paper is very precise. And, last but not least, it is extremely well-written and exhibits a clarity that I never found again.

Unfortunately, the paper failed to influence me.

Michael Rabinovich, AT&T Research, `misha@research.att.com`

[D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy: Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. STOC 1997, 654–663.]

By the time this paper appeared I had been working on the WWW infrastructure for couple years already. At that time, client demand was outpacing server and networks capacity, and techniques to alleviate this problem, such as caching, were actively pursued by researchers, myself included. One particular problem I was considering was locating cached objects in a distributed caching platform. An attractive approach to this problem is to use a globally-known hash function to map an arbitrary URL to one of the caches, so that everyone would immediately know which cache is responsible for which portion of the URL namespace. Unfortunately, with most of demand concentrated among a small number of URLs, this scheme can create serious load disbalance among the caches. Worse, an addition or deletion of even one cache can lead to drastic redistribution of the URL namespace among all the caches, forcing them to go through the warm-up phase again. Karger's et al. paper described a beautiful solution to this problem. Not only did their idea influence my thinking on the immediate problem at hand, but it became one of the standard techniques in my "toolkit" for my other research in the distributed systems area.

The impact of the consistent hashing idea extends beyond my own work, it has proved influential in the general distributed systems research. For example, it provided a foundation for the Chord peer-to-peer network, and through Chord, it has had a significant influence on the p2p research. It was also a starting point for Akamai, one of the few Web infrastructure companies that successfully survived the .com bust. I consider consistent hashing to be one of the most elegant ideas in the distributed systems area, and Karger's et al. paper to be a fine example of theoretical work with a significant practical impact.

Arnon Rosenthal, The MITRE Corporation, `arnie@mitre.org`

[Michael Siegel and Stuart E. Madnick: A Metadata Approach to Resolving Semantic Conflicts. VLDB 1991, 133–145.]

The paper contains two important ideas that have influenced both my research and several MITRE "grand visions" for data sharing. First, it suggested that it is important to capture receivers' interface requirements explicitly, rather than just harvest the queries they issue. The power of this simple idea is discussed in the last paragraph. Second, it provided context rules to describe attribute representations (e.g., simple assertions that the field FuelUnit describes the units for TankerContents, and inferred context: if CUSTOMER.ADDRESS.NATION= "USA" then the CUSTOMER.PRODUCT.PRICE has currency = "US dollar"). Using this sort of knowledge plus libraries of scalar transforms, their group built mediators that generated conversion programs. Over the next decade, I started creating mediators, and ended learning hard lessons about applying such technologies in large enterprises.

We learned that simple context mediation of attributes (when schemas match) was too small a unit of technology insertion. Administrators needed to manually map the context rules to the receiver's schema, an unacceptable burden. Siegel and Madnick's follow-on efforts (Goh et al., ACM TOOIS, 1999) provided a technical solution, but they captured knowledge in an exotic logic known only to a handful of researchers; our customers need a standards-based solution.

Despite these technology transition challenges, the basic idea of context mediation remains compelling. We used it to provide an alternative (simple context rules in OWL) versus current practice of documenting interface semantics in MS Office. We now believe that context mediation is a big win (over hand-coded transforms) if one has structural mappings and if one is mapping many sources to a receiver (or vice versa). We hope that in the near future, high-end integration systems (e.g., IBM's CLIO) will incorporate context mediation.

We also had successes, greatly simplifying several generations of data-sharing visions (large-scale vaporware) for our customers. We applied Siegel and Madnick's first insight (parallel descriptions for sources and receivers) for aspects such as: push versus pull, sending whole tables versus deltas, data quality requirements, data structure encodings, and protocol choice. This work (and also, mediators from MCC and ISI) contributed to the Department of Defense's dropping its vision of a monolithic standard, and managing instead in terms of diverse communities of interest (Rosenthal, et al., SIGMOD Record, December 2004).
