DBMS RESEARCH AT A CROSSROADS: THE VIENNA UPDATE

Michael Stonebraker
Rakesh Agrawal
Umeshwar Dayal
Erich J. Neuhold
Andreas Reuter

Abstract

On April 23, 1993 a panel discussion was held at the IEEE International Conference on Data Engineering in Vienna, Austria, at which five members of the data base research community discussed future research topics in the DBMS area. This paper summarizes the discussion which took place. The panel followed a similar format to that used at Laguna Beach four years earlier, and four of the five panelists attended the earlier conference. As such, we contrast the recommendations of the Laguna Beach participants with those obtained four years later by a similar group.

1. INTRODUCTION

In February 1989, an informal workshop was held in Laguna Beach, California, attended by 7 senior DBMS researchers from Germany and 9 from the USA. This workshop was organized by Erich Neuhold of GMD and Michael Stonebraker of Berkeley, and sponsored by the International Computer Science Institute (ICSI). The purpose of that workshop was to discuss what DBMS topics deserve research attention in the future. During the first day, each participant presented four topics that he was **not** working on that he thought were important and that he would like to investigate. In addition, each participant was asked to present two topics that others were working on, which he thought were unlikely to yield significant research progress. All participants then cast five votes in support of research topics proposed by others. They were also given two votes to indicate agreement with overrated topics. The workshop report [STON89] summarized the discussion which took place, but did not indicate the actual scores from the exercise.

On April 23, 1993, a similar exercise was held in a panel discussion at the IEEE Data Engineering Conference among five participants, four of whom had attended the Laguna Beach workshop. Each panelist was asked to present 4 problems he would like to see solved that he was not working on. Further, he was asked to present 4 problems that he would be happy never to see another paper on. Subsequently, each panelist was given two votes he could cast to support important topics proposed by others and two votes to agree with overrated topics.

Research sponsored by the National Science Foundation Grant RI-91-07455.

The exercise in Vienna was slightly different from Laguna Beach, in that it ensured that "positive" topics could not have "negative" votes and negative topics could not have positive votes. Also, Vienna had a much smaller team of panelists, who did not benefit from an opportunity to discuss the various topics before voting. Even so, the authors believe that contrasting the two sets of scores will provide guidance to the research community in selecting what problems to address.

As such, in Section 2 we briefly review the Laguna Beach scores, followed in Section 3 by the Vienna scores. We close in Section 4 with some comments and views, shared by all five authors.

2. A REVIEW OF LAGUNA BEACH

There were thereby a total 144 "positive" votes and 64 "negative" votes cast at Laguna Beach, and the raw results are summarized in Table 1. Notice that it is possible for some researchers to consider a topic to have much promise and others to consider it having little promise. Hence, the number of positive and negative votes for each topic are presented.

TABLE 1. Laguna Beach Results

Topic	Positive Votes	Negative Votes
End User Interfaces	14	0
Active Data Bases	15	1
Parallelism	11	0
New Transaction Models	10	0
CIM, Image, IR Applications	10	0
CASE Applications	9	0
Security and High Availability	9	0
Large Distributed Data Bases	9	1
DB/OS Interaction	7	0
Transaction Design Tools	7	1
Large System Administration	5	1
Real Time DBMS	3	0
DBMS Implementing Blackboard Paradigm	3	0
IMS-style Joins	3	0
Automatic DB Design	4	2
Tool-kit DBMS Systems	5	3
Data Translation	2	1
OODB	7	6
Dependency Theory	0	3
Interface Between DBMS and Prolog	0	5
New Data Model	0	5
Common OO Data Model	2	7
Traditional Concurrency Control	0	7
Hardware DB machines	0	8
General Recursion Queries	0	10

The thing that amazed the participants was that 10 topics collected 101 of 144 positive votes and six topics received 42 out of 64 possible negative votes.

Essentially all participants wanted to see more research on end-user interfaces to data base systems and active data bases, i.e. rule systems supporting triggers and alerters. Considerable support was also present for parallel query processing on multiprocessor systems, new transaction models, e.g. Sagas [GARC87] and Contracts [WACH92], and finding relevant research problems by studying new application areas such as Computer Integrated Manufacturing (CIM), image data bases, Information Retrieval (IR) applications, and Computer Assisted Software Engineering (CASE). Consideration of high availability, security, scaling problems in very large distributed data bases, interface issues between the DBMS and the operating system and tools to assist users in designing transactions rounded out the list of popular topics.

The six unpopular topics were general recursion, hardware data base machines, exploration of concurrency control schemes supporting serializability on a single machine, a common object-oriented data model, new data models of any kind, and interfaces between a DBMS and Prolog.

General recursion was unpopular because none of the participants had ever seen an application that needed this capability. The participants thought that advocates of general recursion research should either find a credible application for the technology or move on to other more relevant topics. Hardware data base machines were unpopular because the participants felt that software-only data base machines, i.e. conventional multiprocessors running parallel software offered much more promise. Concurrency control was unpopular because of the appearance of a large number of papers on the topic in the mid 1980's, all differing in seemingly minor ways. The participants thought that little progress would be made by continuing to publish minor variations on a set of common themes. Data models (object-oriented or otherwise) were not in favor, because the participants had seen a large number of them, differing in small ways, and they did not want to see any more. Lastly, interfaces to Prolog were not considered the best way to build expert data base systems. Rather rules systems integrated with the DBMS should be the focus of research activity.

It is an understatement to say that the report was immediately controversial. Perhaps the biggest problem with the report was that the composition of the participants was primarily from the systems area. For example, there was nobody from the theory community, and the representative from the deductive DBMS community was forced to cancel at the last minute. Hence, the participants did not represent a broad cross section of DBMS researchers. As such, their collective judgement may be biased in assorted ways.

3. THE VIENNA UPDATE

In this section we present the scores captured during the Vienna panel discussion. Although opinions were hastily conceived and from a narrower collection of researchers; nevertheless some of the conclusions that can be drawn are very interesting. There were a total of 30 positive votes and 30 negative votes cast by the panelists, and we summarize the results in Table 2.

There was near-universality of interest in five topics. The panelists were enthusiastic about new user interfaces, such as workflow languages and collaboration tools. They lamented that progress in this area continues to be done by industry, and the research community has very little impact on this important topic.

In addition, there was interest in studying problems of scaling DBMSs to very big (multi-terabyte) data bases and very big (thousands of clients) systems. Scaling to terabyte data bases entails coping with memory management in a multilevel hierarchy and performing very intelligent caching. Also, a DBMS must cope with millions (and perhaps billions) of objects. Scaling to a large number of clients requires solving such matters as installing a new copy of an

TABLE 2. Vienna Results

Topic	Positive Votes	Negative Votes
End User Interfaces	4.5	0
Big Systems	4.5	0
Legacy Applications	4	0
Multimedia Applications	4	0
Data Mining	4	0
Mobile Data Bases	2	0
Method Optimization	2	0
Embedded DBMS	2	0
Object Repositories	1	0
OO Data Base Design	1	0
Constraints	1	0
Relational Extensions	0	1
Synchronization theory	0	1
Simplistic Data Integration	0	1
Less than ACID Transactions	0	2
Persistent C++	0	2
Multi-data base Transactions	0	3
New OO Data Models	0	3
Replication Algorithms	0	4
Bizarre Performance Studies	0	4
Traditional Engines	0	4
General recursion	0	5

application program without taking the system down and keeping track of an application in which certain clients are running different versions of the same application.

Furthermore, essentially all large companies are running their business on large application systems that are at least ten years old. Such systems are typically poorly structured and often use obsolete DBMS technology or even no DBMS at all. Managers in these companies want to retire this "legacy" code to move to modern DBMS and client-server technology, and they can proceed by a global rewrite or incremental migration. Since total rewrites are perilous and failure prone, users need help in generating feasible incremental migration strategies for legacy applications. Participants were enthusiastic about reverse engineering techniques as well as architectural suggestions such as [BROD93].

Problems associated with storing large multi-media objects in data bases also stood out. These include how to build indexes on the content of such objects and how to provide services such as guaranteed delivery.

Lastly, "data mining" was a very popular topic. It is motivated by the decision support problem faced by most large retail chains. They record every item that is purchased in every store by capturing this data at the point of sale. Buyers and merchandise arrangers use this data base to rotate stock and make purchasing decisions. The query to such a data base is "tell me something interesting". Specifically, users want a system that would "mine" the data for useful information.

Three other topics received lesser support. The first concerned problems in mobile data bases. There will be applications where clients have hand-held devices, which may only be intermittantly connected to a DBMS server. As such, system designers must cope with issues such as operating the system in disconnected mode and then performing downstream version merging. Furthermore, optimizing queries to minimize power consumption is a worthwhile exercise.

Second, extendible and object-oriented data base systems allow user to add user-defined functions to the DBMS. Such functions can be written in the query language, or they can be written in a general purpose programming language such as C with intermixed SQL queries internal to the function. In this latter case, the function is "opaque" and its cost of execution cannot be identified. How to make a query optimizer intelligently deal with queries containing such functions was thought important by some panelists.

The final topic was one of embedded DBMSs. Here, the focus was on applications such as telephone switches where the hardware and software are a "closed world" which is constructed at the factory and not changed in the field. There is no requirement to run arbitrary user programs or for protecting the DBMS from application programs. Instead other issues arise such as a very high availability requirement, which requires that new versions of the software be installable without taking down the device, and extremely high performance requirements.

The panelists were uniformly hostile to general recursion. Ever increasing numbers of papers are being written to define yet another declarative semantics of stratefied aggregation/negation or provide a twist on magic set optimization techniques. It has been four years since Laguna Beach, and there is **still** no known user of this technology. The segment of the DBMS research community that writes papers on this topic should **really** be charged with finding applications that can use their results.

The optimization of traditional single site DBMSs for business data processing applications was also poorly received. The sense of the panelists was that this topic is well understood and that we should declare it a solved problem. Future research will be incremental in this area, and researchers are "polishing a round ball". In fact, one panelist observed that in 1995 it will be possible to buy 1000 transactions per second for $100,000, and that there are only two known applications which require higher transaction rates. More typical applications require 100 transactions per second and will need only a small portion of a cheap machine. They will increasingly be able to get by with "brute force" solutions to their DBMS problems, rendering further research in this area of questionable value. In a sense, Laguna Beach declared traditional concurrency control a dead topic; Vienna extended this death warrant to single site DBMSs.

The third topic received with disfavor was performance studies of artificial environments. Two examples were heavily cited. First, studies of "toy" disk-based data bases of a few megabytes or less which were fronted by even smaller main memory caches were scorned. Current workstations routinely come with 32 Mbytes of memory, and performance studies that assume a cache size considerably smaller than this number seemed unreal to the panelists. In effect any study with a small number of Mbytes of data that did not assume full main memory caching seemed silly. Hence, performance studies should use technology parameters reflecting current reality, not some past reality.

Second, Jim Gray once formulated the following law:

> In a well-designed data base, the probability of waiting as a result of a lock request is less than 0.01

Specifically, in a real data base, transactions rarely wait. The reason is that an application cannot afford to have humans sitting idle waiting for lock releases. A data base administrator faced with this situation will redesign his data base to make the probability of waiting very rare.

There have been a **large** number of papers recently published in which the probability of waiting is more than one order of magnitude higher than that in Jim Gray’s law. Such authors should realize they are optimizing a DBMS for a load that will never be experienced in the real world.

Another unappealing area was algorithms for updating multiple copies of objects in distributed data bases. There have been a **large** number of papers exploring new techniques in this area, and the panelists felt that a large number of additional papers could be written. Moreover, most entail setting read locks at more than one site and write locks at less than all sites. Quorum and majority consensus algorithms have this property. However, a read command need be sent to only one site, and a single lock request to this site can be piggy-backed onto the query. Similarly, all copies must be updated and a lock request again can be piggy-backed. As such, it is difficult to beat a scheme that reads and locks any copy and writes and locks all copies that are currently operational, as explained in [ELAB85]. Any author writing a replication paper must keep in mind this simple fact.

Another hostilely received topic was any additional object-oriented data models. The feeling was that **lots** of models have been invented, and lots more will presumably be discovered, all differing in minor ways. The feeling of the panelists was that no more papers should be written in this area. This reinforces the same feeling from the Laguna Beach participants.

Another topic with 3 negative votes was transactions spanning multiple data bases. Specifically, the panelists were negative on algorithms supporting two-phase commit in a heterogeneous distributed data base environment in which the various local DBMSs do not support a **prepare** message. This requires sophisticated and expensive algorithms to simulate this capability outside the DBMS. The feeling of the panelists was that XA would force all vendors of data managers to support a common distributed transaction capability, and render this problem irrelevant. In this scenario, a prepare message would only be missing from legacy "home brew" systems, and it is unlikely that the sophistication to implement simulations of two-phase commit would be present in such application shops. Hence, this problem is not relevant to any real world situation.

Two final topics deserve comment. First, the panelists felt that there is a limited commercial market for persistent C++ data base systems. Compared to the SQL DBMS market, persistent C++ is perhaps 1-2%, and that it is not likely to "take off" in the near future. As such, researchers should focus their energy on problem areas with a higher possibility of product penetration.

The last topic was the study of concurrency control and crash recovery systems that supported less than ACID properties. The feeling was that such schemes are not general purpose enough to ever find much favor in real DBMSs. Hence, there is limited applicability for such research.

4. COMMENTS ON THE EXERCISE

One striking feature of these two exercises is that most of the important Laguna Beach topics have been extensively addressed in the intervening four years, and have dropped off the list of things requiring attention. Only user interfaces remains on the 1993 list. In addition, work on the negative topics has largely ceased. With the exception of general recursion, most of the negative topics have disappeared from the 1993 list. As a result, it appears that the DBMS community has largely addressed the opinions voiced in the Laguna Beach report.

However, we must now ponder the Vienna results in Table 2. One member of the Vienna audience pointed out that the 1993 Data Engineering conference had a large number of papers in the areas considered negative by the panel and almost no papers in areas considered important. Assuming that this conference is typical of DBMS research, it appears that our community is largely working on the wrong problems.

Put more strongly, we are at a **crossroads,** in that the traditional topics we have studied such as buffer management, concurrency control and query optimization should be declared "solved". However, as a community we continue to "plow" the familiar ground and we appear to be increasingly "polishing a round ball". On the other hand, the problems identified by the panel as important have the property that they are both very hard (e.g. data mining is almost certainly "AI complete") and also away from the center of previous DBMS activity. Hence, our community is at a crossroads where we can either continue along the traditional road or take a path exploring largely unknown terrain.

It is, of course, safer to take the traditional path. Program committees react favorably to "more of the same", and often react badly to papers in new areas, especially if they do not contain well thought out formal results. As a community, we should consciously break out of this mold.

Another way of considering the 1993 table is that DBMS research in the 1990's should have an application focus rather than a technology focus. The old wisdom was to find a technical problem and then solve it, while the new adage appears to be "find a customer" and then solve the problem that he explains to you. The important problems from the 1993 table appear to have largely come from applying this advice.

A last lament echoed in the halls of the conference but not directly by the data in Table 2 is that there are simply too many papers being published. The number of researchers needing to gain tenure as well as the number of conferences has increased dramatically. It is now nearly impossible to keep up with the DBMS literature across more than a very narrow slice of the research terrain. Moreover, most researchers seem to dissect their ideas into "least publishable units", so as to maximize the length of their vitae, contributing further to the paper explosion. A way to lower the number of words published is clearly needed.

REFERENCES

[BROD93] Brodie, M. and Stonebraker, M., "Incremental Migration of Legacy Data Base Applications," GTE Laboratories, Waltham, Mass., Technical Report 93-12, January 1993.

[ELAB85] El Abbadi, A. et. al., "An Efficient Fault-tolerant Protocol for Replicated Data Management," Proc. 1985 SIGACT-SIGMOD Symposium on Principles of Data Base Systems, 1985.

[GARC87] Garcia-Molina, H. and Salem, K., "SAGAS," Proc. 1987 ACM-SIGMOD Conference on Management of Data, San Francisco, Ca., May 1987.

[STON89] Stonebraker, M. and Neuhold, E., "The Laguna Beach Report," International Institute of Computer Science Technical Report #1, Berkeley, Ca., June 1989.

[WACH92] Wachter, H. and Reuter, A., "The ConTract Model," in "Transaction Models for Advanced Database Applications," Morgan-Kaufman Publishers, Redwood City, Ca., 1992.