Research Directions in Data Base Management Systems* +

by D. Tsichritzis

Computer Systems Research Group
University of Toronto
Toronto M5S 1A4, Canada

1. Introduction

Data Base Management Systems (DBMS's) have become "in vogue" recently. The area is not only fashionable but also very important. In some disciplines the trends of research rest purely on the taste of the researchers, or the research agencies. However, the directions in DBMS's come straight from the marketplace. The area was originally developed and is heavily influenced by the problems large organizations have to manage effectively and efficiently large volumes of data. There is no select group of individuals in DBMS who collectively decide what will be the next problem to attack or the next project to start. The users have problems which influence the manufacturers to propose solutions which, in turn, guide the researchers in their research activity. Periodically large user groups or associations collect opinions on the state of the art and what needs to be done next. For instance, Guide Share, CODASYL DBTG, NBS and ANSI have reports about DBMS's [1,13,25].

*In this paper we take a candid view of DBMS research. Let us hope that nobody gets upset.

+ This is an invited paper at the Sixth Conference on Information Sciences and Systems, John Hopkins University, 1977.

Since DBMS is a relatively new discipline many people have converged into it from other areas. For instance, one can find DBMS experts with backgrounds in programming languages, operating systems, artificial intelligence, theory of computing, etc. People are still moving in. The main reason for this intellectual "gold rush" is that researchers expect a very high return for their investment. They feel that the area is rather unsophisticated, the problems rather simple and the solutions easy to come by. In our opinion, this is no longer true. The area has developed a large body of knowledge and a large number of published documents. Granted, that most of the papers are not worth reading. There are many papers, however, that are very important. Hence, the area is not so easy to get into. One can point at major contributions a few years back which, in retrospect, are rather simple. At the time they were great advances. There are still simple unimportant problems and hard important problems to be solved. However, we doubt whether anybody will find a simple very important problem in the future. Almost all the simple important problems have been solved. Hence, there is the danger that we are going to be deluged with a great number of solutions to non-existant problems. A problem is non-existant if nobody really cares about its solution.

To illustrate how fast the area has evolved we will outline as examples two problem areas. They were very important research directions a few years ago. However, we feel that future contributions will be evolutionary rather than revolutionary.

A few years ago some DBMS approaches and techniques were at such a rudimentary stage that people were building prototype systems to investigate them [3,38]. It will be very hard to surpass the efforts of the past unless one has enough resources to make a prototype system realistic [27]. But this implies an organization and not 3-5 people part-time. A prototype system which can be built by 3-5 people over 2 years (the standard university project) will probably be mainly educational and not research oriented. Nobody is interested in "here is the way we did it" type of papers anymore. We know that DBMS's can be built. To build them better and to demonstrate the advantages cannot be done anymore as a "shoe string" operation.

A few years ago individual researchers made major contributions in data models and data languages [14]. Many others have followed suit and have proposed a plethora of data models [29]. However, we feel that now is a time for consolidation rather then innovation. People are not interested anymore in new data models although they are still interested in results about data models (which are hard to come by). Most "new" data models are, anyway, combinations of ideas from the "oldies". Coexistence efforts are also very important, e.g., multi-model data language interfaces [20]. A "new" data model or data language with no demonstrable advantages is not a major contribution anymore.

In the rest of the paper we outline the areas that, in our opinion, are still worth investigating. We expect the major research results to come from these areas. The division of the areas is rather arbitrary and the names may be even more confusing. However, we hope to give a feeling of our projected evolution of DBMS research.

2. Multi-purpose architectures

One of the goals of DBMS's is to provide a flexible interface to the users. In this way user programs are "independent" from the physical placement of the data. In addition, it would be nice if user programs are "independent" from the logical structure of the data. In this way some of the logical relationships may evolve while the user only sees his own view of data. One way of achieving this logical data independence is through a multiple level architecture. A framework for such an architecture has been proposed by ANSI/X3/SPARC [1]. However, there is still considerable debate about some of the features of such an architecture. Typical questions that are being asked are:

1) What is the nature of the conceptual schema?
2) How will multiple model external views be supported?
3) Is an ANSI/X3/SPARC type of architecture realistic?
4) Which of the proposed interfaces should be standarized?
5) How will integrity constraints be enforced among external schemata?
6) Will an ANSI/X3/SPARC type of architecture solve the conversion problem by enabling DBMS's to evolve?

In essence, the main problem is the future architecture of DBMS's. Should they provide more versatile and flexible features? How should they do it? The possibility of an imminent standarization gives particular urgency to these problems. As a byproduct of this effort much work has already been done on the definition of concepts relating to DBMS's [1]. We do not expect a radical quantum jump in any direction in this problem area. It will take some time to reach any conclusions. Experience in software system architecture is very important to do any research in this area.

3. <u>Performance analysis and improvement</u>

DBMS's use some standard mechanisms to store and access data, e.g., files, inverted files, transposed files, etc. The organization of these mechanisms follows well known techniques, e.g., B-trees, binary trees, hashing, etc. The problem comes when data volumes get large. It is not a question of doing something but doing it well. An inefficient algorithm or access method can severely limit a DBMS's performance. There is a need for requirements analysis, modeling, evaluation, and performance studies. Typical problems of this nature are:

1) Which secondary indices should be kept?
2) What mechanism should be used to implement secondary indices?
3) Should other access paths, separate from secondary indices, be kept and how?
4) How can one cluster data which is accessed as a unit?
5) Data encryption and compression.
6) Hardware solutions for fast searching.
7) Operating system tuning for DBMS requirements.
8) Performance indicators and measurement.
9) Level of locking for concurrency.
10) Proper utilization of different storage media.
11) Optimal query processing.

The analytic models for some of these problems can become quite complicated. Experimental results are very hard to come by. It is not easy to find an available operational DBMS with which to experiment. In addition, DBMS's are sufficiently sensitive that DBA's are very reluctant to let anybody "play" with them.

4. Distributed Data Bases

Computer networks have evolved sufficiently for distributed DBMS's. Network data transmission is fairly efficient and inexpensive. In addition, there are some data bases that are naturally distributed. It is obvious that distributed data base management systems will be built and operated successfully in the future. Research, however, is lagging behind commercial development. There is much research work in network data transmission, in file distribution in the network, in heterogeneous network protocols and in large data repositories for the network. However, all of this work is not directly related to distributed DBMS's. The real proplem is to develop techniques that provide data independence of the user from the idiosynchracies of a data base distributed in an homogeneous computer network. Some of the associated problems are:

1) Schema distribution
2) Data placement
3) Message-oriented system architecture
4) Nature of a transaction in a network
5) Enforcing the integrity of the distributed data base
6) Controlled redundancy
7) Concurrent updates and their meaning.
8) Recovery in a distributed data base.

For some time the emphasis in computer networks was on data transmission. At this point, it is important to look carefully at the nature of the data being transmitted. In distributed data bases it is imperative to transmit data among the nodes efficiently. However, it is also important to have a system wide control which assures the correctness of the operation.

5. Data types

Programming language experts have aknowledged finally the importance of data. Their answer in a capsule is: data types. However, programming language experts and DBMS experts have refused until now to take each other seriously. There are still programming language persons who believe that a data base is nothing more than a "complicated" data type. On the other hand, some DBMS persons dismiss data types as "blue sky nonsense". The truth is, of course, somewhere in the middle. Some contributions on data types are obviously important for DBMS's [24]. On the other hand, we are very far from an axiomatization of a data base. There are the differences between data types and DBMS problems. The gap has to be bridged and any contributions are welcome. Typical problem areas are:

1) Data types which overlap in terms of data elements.

2) Formalization in terms of data types of some of the common DBMS structures, e.g., DBTG set type.

3) Lattice of data types, which reflect a DBMS, and their properties.

4) Axiom enforcement in the implementation of a data type.

5) Concurrent sharing of common elements of different data types.

6) Feasibility study concerning the application of data types in the resolution of any "reasonable" DBMS problem.

It may be that data types can provide a rigorous framework much needed in DBMS's. However, it may also be that data types are too cumbersome to use in any meaningful way. Time will tell.

6. Natural language interfaces

For some time there has been a "rapprochement" between Artificial Intelligence and DBMS's. The DBMS architects want systems which provide casual user interfaces [15]. On the other hand, Artificial Intelligence persons want to apply their techniques in useful and realistic systems [45]. As a first step Question-Answering systems are being fitted into data base applications. In addition, Artificial Intelligence models are being proposed as a solution to data base semantic problems [35]. The area is very exciting. It is not exactly clear that DBMS's need natural language interfaces. However, they definitely need some of the techniques used to handle natural languages. Any model or technique which can handle an interface as complicated as natural language can be toned down to handle some of the mundane semantics in DBMS's. However, exchanging generality for effectiveness and efficiency is not always easy. Typical problems that need investigation are:

1) Semantic description
2) Sophisticated data base dictionaries
3) Mapping semantics into DBMS schemas
4) Enforcement of semantic constraints
5) Handling of free-format casual user interfaces

6) User-DBMS dialogue

7) Building semantics in data models

8) Problem specification languages

9) Infological models [39].

7. Concluding remarks

In the previous paragraphs we mentioned a few areas in DBMS which need investigation. Our list is by no means exhaustive. It is important to notice that all proposed areas are essentially team efforts between DBMS and an established discipline. We have the correspondence.

Multi-purpose architectures	DBMS - Software Engineering
Performance improvement	DBMS - Performance Analysis
Distributed data bases	DBMS - Computer Networks
Data Types	DBMS - Programming Languages
Natural Language interfaces	DBMS - Artificial Intelligence

Theoretical problems have already been investigated in the associated areas we mention, e.g., Programming Languages, etc. If we extrapolate from the past, it seems that the combination areas with DBMS's can be fruitful for theoretical study. Specifically, work on data model properties like normal forms and functional dependencies is very important to derive proper conceptual schemas [7]. Queueing Networks can probably be adopted for performance analysis of DBMS's. General synchronization models like Petri nets, should be relevant in analysis of message networks of distributed data bases. Theorem proving techniques are relevant to abstract data types and their axiom specifications. Finally, formal semantics should be used to clean the data model semantics in both AI and DBMS's. It seems, therefore, that most standard theoretical areas can be adopted to DBMS study. It will only take an effort from the theoreticians to understand the issues in DBMS's.

We propose the following steps for a potential researcher in DBMS.

1) Read a book about DBMS to get the overall picture [19,30,43].

2) Read some of the key classical papers, e.g., Codd, Bachman, etc.

3) Concentrate in an area where your background will give you the necessary tools. The correspondence table given above should help.

4) Read seven (7) papers in your area of interest. Ask for a recommendation of which papers to read.

5) THINK

6) Try out your results on somebody who has a wide, and if possible pragmatic, experience in DBMS's.

7) Add your gem to the already unlimited collection of DBMS litterature.

References

1) ANSI/X3/SPARC [1975]. "Intrim Report NASI/X3/SPARC Study Group on Data Base Management Systems," FDT7(2).

2) Ashenhurst, R.[1974]. "A Great Debate," CACM17,360. Ashenhurst, R.L., and Vonderohe, R.H.[1975]. "A Hierarchical Network," Datamation 21,(2)40-44.

3) Astrahan, M.M., et al.[1976]. "System R: Relational Approach to Database Management," ACM Trans. Database Sys. 1(2),97-.

4) Bachman, C.W.[1969]. "Data Structure Diagrams," Data Base 1(2), 4-10.

5) Bachman, C.W.[1973b]. "The Programmer as Navigator," CACM 16,653-658.

6) Bachman, C.W.[1974b]. "Summary of Current Work: ANSI/X3/SPARC/Study Group-Database Systems," FDT 6(3), 16-39.

7) Bernstein, P.A.[1976]. "Synthesizing Third Normal Form. Relations from Functional Dependencies," ACM TODS Vol.1, No.4, Dec. 1976.

8) Boyce, R.F., Chamberlin, D.D., King, W.F.,III, and Hammer, M.M.[1974]. "Specifying Queries as Relational Expressions," in Data Base Management (Klimbie, J.W., and Koffeman, K.L., eds.), pp.169-176. North-Holland, Amsterdam.

9) Chamberlin, D.D.[1976]. "Relational Data-Base Management Systems," | ACM Comput. Surveys 8, 43-66.

10) Chamberlin, D.D., and Boyce, R.F.[1974]. "SEQUEL: A Structured English Query Language," Proc. ACM SIGMOD Workshop on Data Description, Access and Control, pp.249-264.

11) Chen, P.P.-S.[1976]. "The Entity-Relationship Model: Toward a Unified View of Data," ACM Trans. Database Sys. 1 (1)9-36.

12) Childs, D.L.[1968]. "Feasibility of a Set-Theoretic Data Structure-A General Structured Based on a Reconstituted Definition of Relation," Proc. IFIP Congr. 1968, pp.162-172. North-Holland, Amsterdam.

13) CODASYL DBTG[1971]. CODASYL Data Base Task Group Report, Conf. Data Sys. Languages, ACM, New York.

14) Codd, E.F.[1970]. "A Relational Model of Data for Large Shared Data Banks," CACM 13,377-387.

15) Codd, E.F.[1974a]. "Seven Steps to Rendezvous with the Casual User," in Data Base Management (Klimbie, J.W., and Koffeman, K.L., ed.), pp.179-199. North-Holland, Amsterdam.

16) Codd, E.F.[1974b]. "Recent Investigations in Relational Data Base Systems," Proc. IFIP Congr. 1974, pp.1017-1021. North-Holland, Amsterdsam.

17) Codd, E.F., and Date, C.J.[1974]. "Interactive Support for Non-Programmers: The Relational and Network ApproAches," Proc. ACM SIGMOD, Data Models: Data-Structure-Set versus Relational (Rustin,R.,ed.), pp.11-41.

18) Cullinane Corp.[1975]. Integrated Database Management System (IDMS) publications: Data Definition Languages, Utilities and GCI Reference Guide, release 3.1.; Data Manipulation Language Programmer's Reference Guide, release 3.1.

19) Date, C.J.[1975]. An Introduction to Database Systems, Addison-Wesley, Reading, Massachusetts.

20) Date, C.J.[1976]. "An Architecture for High-Level Language Database Extensions," Proc. ACM SIGMOD, pp.101-122.

21) Date, C.J., and Codd, E.F.[1974]. "The Relational and Network Approaches: Comparison of the Application Programming Interfaces," Proc. ACM SIGMOD, Data Models: Data-Structure-Set versus Relational (Rustin, R.,ed.), pp.83-113.

22) Engles, R.W.[1972]. "A Tutorial on Data-Base Organization," Annual Rev. Automat. Programming 7,1-64.

23) Fry, J.P., and Sibley, E.H.[1976]. "Evolution of Data-Base Management Systems," ACM Comput. Surveys 8, 7-42.

24) Guttag, J.[1975]. "The Specification and Application to Programming of Abstract Data Types," CSRG Tech. Report #59, University of Toronto, Sept. 1975.

25) Guide-Share Data Base Management System Requirements[1970]. Joint Guide-Share Data Base Requirements Group, New York.

26) Hammer, M.M., and McLeod, D.J.[1975]. "Semantic Integrity in a Relational Data Base System," Proc. ACM Int. Conf. Very Large Data Bases (Kerr, D.S., ed.), pp.25-47.

27) Hardgrave, W/T.[1972]. Theoretical Aspects of Boolean Operations on Tree Structures and Implications for Generalized Data Management. TSN-26, Computation Center, Univ. of Texas at Austin.

28) IBM[1975]. Information Management System/Virtual Storage (IMS/VS) publications: General InformATION MANUAL, GH20-1260-3.

29) Kerschberg, L., Klug, A., and Tsichritzis, D.C.[1976]. "A Taxonomy of Data Models," in Systems for Large Data Bases (Lockemann, P.C., and Neuhold, E.J., eds.), pp.43-64. North-Holland, Amsterdam.

30) Martin, J.T.[1975]. Computer Data-Base Organization. Prentice-Hall, Englewood Cliffs, New Jersey.

31) Michaels, A.S., Mittman, B., and Carlson, C.R.[1976]. "A Comparison of the Relational and CODASYL Approaches to Data-Base Management," ACM Comput. Surveys 8, 125-151.

32) MRI Systems Corp.[1972]. SYSTEM 2000 publications: General Information Manual, Basic Reference Manual, Immediate Access Feature, MRI Systems Corp., Austin, Texas.

33) Mylopoulos, J., Schuster, S.A., and Tsichritzis, D.C.[1975]. "A Multi-Level Relational System," Proc. AFIPS 44, NCC, 403-408.

34) Ozkarahan, E.A., Schuster, S.A., and Smith, K.C.[1975]. "RAP: An AssociAtive Processor for Data Base Management," Proc. AFIPS 44, NCC, 379-387.

35) Roussopoulos, N., and Mylopoulos, J.[1975]. "Using Semantic Network for Data Base Management," Proc. ACM Int. Conf. Very Large Data Bases (Kerr, D.S., ed.), pp.144-172.

36) Software AG[1971]. ADABAS publications: General Information Manual, Reference Manual, Utilities Manual. Software AG, West Germany.

37) Steel, T.B.[1975a]. "Data Base Standardization: A Status Report," in Data Base Description (Douque, B.C., and Nijssen, G.M., eds.), pp.183-195. North-Holland, Amsterdam.

38) Stonebraker, M.R., Wong, E., and Kreps, P.[1976]. "The Design and Implementation of INGRES," ACM Trans. Database Sys. 1(3), 189-222.

39) Sundgren, B.[1974]. "Conceptual Foundation of the Infological Approach to Data Bases," in Data Base Management (Klimbie,J.W., and Koffeman,K.L.,eds.), pp.000-000. North-Holland, Amsterdam.

40) Taylor, R.W., and Frank,R.L.[1976]. "CODASYL Data-Base Management Systems," ACM Comput. Surveys 8, 67-103.

41) Tsichritzis, D.C.[1976]. "LSL: A Link and Selector Language," Proc. ACM SIGMOD, pp.123-133.

42) Tsichritzis, D.C., and Lochovsky, F.H.[1976]. "Hierarchical Data-Base Management: A Survey," ACM Comput. Surveys 8, 105-123.

43) Tsichritzis, D.C., and Lochovsky, F.H.[1977]. "Data Base Management Systems," Academic Press, New York.

44) Wong, H., and Mylopoulos, J. "Two Views of Data Semantics: A survey of Data Models in Artificial Intelligence and Data Management", unpublished manuscript.

45) Woods, W.A.[1973]. "Progress in Natural Language Understanding-An Application to Lunar Geology," Proc. AFIPS 42, NCC, 441-450.

46) Yormark, N.[1976]. "The ANSI/X3/SPARC/SGDBMS Architecture," SHARE XLIV, DBMS Conference, Montreal, North-Holland 1977, editor, D. Jardine.