Pragmatics of Database Management
Outline of the Panel Discussion

Panel Chair

Edgar H Sibley,
George Mason University

Panel Members

Matthias Jarke,
New York University

Cecil S McMinn,
Upjohn Pharmaceuticals

John Murray,
Immigration and Naturalization Service

Randall Rustin,
Algorithmics, Inc

Ken Sloan,
Applied Data Research

1 0 General Aspects -- E H Sibley

Database Management has, at last, been accepted as an important part of the Information Systems World DBMS are accepted as useful tools in much of the information industry We no longer need to market them as hard as we did in the late sixties, we need not apologize for the terrible DBMS-based fiascos of the "misapplications" during the seventies We can admit that the problems were often due to enthusiastic misuse and mismanagement of the DBMS Sometimes the mistake was in not creating a Data Administration (DA) group or in creating one without transferring any authority to it But there are still some old and new problems Often due to lack of adequate funding and suitably educated personnel (and this also speaks of poor management), new problems are sometimes t'e old in a new guise

DBMS applications are still changing There are more and more users of Relational DBMS (though hierarchic and network models are still the prime favorites for big systems) Problems of large databases during the past decade are being replaced by ones of "enormous" databases in the present and immediate future, probably with Engineering Design and Production Systems leading the way CAD/CAM systems and Programming Support Environments with software configuration management and requirements change control are obvious examples Also there is a growing movement towards distributed information systems, where the control of the whole network of databases relies on the use of DBMSs as building blocks

When assessing the value of an information system, users are the only ones who can provide the true measure In designing, implementing, and managing the operation of a system (with or without a DBMS), problems range from organizational (political) to useless outputs (poor designs) and slow speeds (poor techniques). Problems are compounded when there is need for a large database with a complex structure There is also a tendency for "rougher" politics as the information systems department, due to size and budget, achieves higher organizational visibility

This sets the stage for today's DBMS problems, but there are new factors in our future First, there is the addition of viable DBMSs for "minis and micros" Although these DBMSs,with associated spreadsheets and other software, are being bought and "tried out" in industry, they are seldom integrated into the total information system plan But they are undoubtedly very popular today, indeed, it seems that we could be embarking on an "anarchism" involving a proliferation of incompatible databases Are we about to revisit the horrors and disasters of the sixties and early seventies? Yet micro-DBMSs offer hope for good and inexpensive implementation of distributed systems Effective use of them in organizations seems possible, with planned orchestration of redundancies and some temporary inconsistencies in the partially or totally replicated databases

A second factor that could affect the future is in the increases "potential" of A I (artificial intelligence) techniques These may soon be married to older DBMS technology to produce "semi-structured systems" and provide a bridge from data to information and thus to knowledge

Some questions of today and the future therefore are

- Is the current DBMS suitable for the wide set of problems we would like to implement today?
- What are today's practical problems?
- How can we transition to future systems without high cost?

- What will make really big databases (10 billion characters up) "machineable"?
- What will such large systems users expect?
- Will users trust the larger systems?
- How will the micro/mini-DBMSs affect the design of both the "coordinating" or controlling DBMSs and the application I/S?
- Will the changes be radical?
- Will future systems be mainly unstructured?
- If so, how will they be accessed effectively? (both from a user and machine standpoint)
- Is the future in Expert/Knowledge-Based Systems, or are they just extensions of DBMSs?
- Will such systems be defined and constructed to make the classification of its knowledge available to both the system and user?

2 0 Panelists' Viewpoints

Matthias Jarke points to the tremendous growth in the field of microcomputers There are many DBMSs for micros, but our community seems untouched by their potential The first micro-DBMS were used in limited capability, more recently, organizations have recognized the problems resulting from the loss of a consistent, shared data resource Micro-mainframe integration, expecially designed to share and exchange data, is one of the "hot topics" in business data processing today

This provides a major challenge for practical database research

> How can we combine the efficiency and privacy aspects of PC workstations with the fast and convenient access and organizational consistency of a central database?

Specific research issues include

- Optimal allocation of data and worktasks between local and centralized systems,
- The design of special communications protocols for use at the application-level (e g , for CAD operations),
- Support for the robust evolution of large-scale PC networks caused by continuous introduction of new data models, DBMS, and uses, and ways to utilize advances in general and special-purpose hardware

Furthermore, application areas for such large-scale distributed databases often have their own special set of requirements, and the ramifications for the database design seem little understood, as an example, consider the case of office and clerical work, or the needs in data analysis, and finally the general purpose Decision Support System

A research strategy could proceed in two ways

> First, on the general problems of distributed, heterogeneous large databases, and
> Second, in specialization of large distributed database systems for unusual applications

Cecil McMinn first contrasts the two DP centers at Upjohn, one (corporate) dealing with production, sales, and other financial matters using IMS as its workhorse, and the other (in the Research Center) a relational one which began using System R in late 1977 with data generated in the Pharmaceutical R&D Division R&D later transferred over to SQL/DS for the VM operating system

Next, concentrating on R&D, he notes that the small DP staff did not have a D A function prior to the adoption of SQL/DS, and many user databases were developed by programmers for their users This became a problem, with lack of data sharing causing a condition akin to that prior to the introduction of the DBMS, fortunately this situation has now been corrected Usage has increased "Old-timers" (originally afraid to use a terminal) saw how easy it was to obtain data and realized their actions could not "destroy the computer", they have become its best advertisers and want all their data to be available via the DBMS

The management of the consequently large number of databases then became a problem, because the use of a DBMS was not always appropriate The DP staff and DA function were forced to be selective in incorporating new applications And this meant more planning and up-front management Lack of efficient response due to badly formulated (large) queries and the effect on others required education in query language statement formulation --especially when the databases were used by almost all employees in the complex Nowadays, applications run the gamut, including such functions as inventory update and control, chemical structures, biological data, and conference room scheduling

John Murray asks whether the inherent design of today's DBMS takes advantage of fourth generation hardware capabilities The vast majority of the present "well-tuned and efficient" systems were developed in the mid-seventies with the objective of helping users by providing

- A Data definition facility,
- Data dictionary interfaces, with cross referencing,
- Easy access to primary data and any related data, whether physically contiguous or not,
- Reduction of redundancy,
- Fourth generation program development tools,
- Security of data at the lowest level, etc

The problems of implementing a DBMS for this set of needs often resulted in inefficiencies, or a slow response, or both Thus DBMS designers had to work round the hardware shortcomings Therefore, present DBMSs were designed to overcome limitations of old hardware architectures, and though these may not now apply, the working DBMS cannot be altered to take advantage of the newer hardware For example, the DBMS today allows the "advantage" of breaking up the database into small logical and physical segments and then creating many files and pointers Does this stem from the fact that I/O buffers and track size used to have to be small? (This is now as silly as trying to save core)

Hardware technology, by the nineties, will allow more "intelligence" in the mainframe and control units, this will require movement of larger volumes of data and larger buffers But this trend is almost here, disk buffer sizes have moved from eight or sixteen KB to a recent announcement of two to 64 MB But how will today's DBMS take advantage of this? Is it possible to reverse the older policies or are we to stay old fashioned?

Ken Sloan believes that every new technology is dominated by a few ideas and their implementation in particular forms, e g , automotive technology has the internal combustion engine and four wheels as the dominating components Early DP systems were built to solve numeric or counting operations on identifiers, and these implementations proved so good that the representation of numbers defined the approach to that used for all symbols in general

The evolution of any technology emphasizes initial ideas and forms more and more heavily as attempts are made to solve both the initial problem and then the new or unforeseen ones Widespread use and acceptance of the original ideas make it unlikely that other forms are even considered For automotive technology, recent emphasis in problems of pollution, fuel economy, and reliability have been difficult to solve within the old constraints of the original and now entrenched forms For DP, the most serious current problem is in accomodating the need for better processing of information that does not fit within the 'humeric paradigm"

With hindsight, we see that the automation and management of primarily numeric values was only the first phase of an ongoing many-phased evolution towards coherent abstractions of the real world

The automated model of the important aspects of the problem should allow better understanding of the abstraction, its easier manipulation, and good communication about the "real world" it represents

- How will these challenges be met?
- Will DBMS continue to ignore free form text and its effective retrieval?
- Will needs of engineering CAD/CAM systems continue to be ignored by DBMS builders
- Will semantic representations (now being developed by AI researchers) be reflected soon in commercial DBMS?
- With these new generation DBMS available, how can they coexist with older systems?

The practical problems today lead to the question

> Can we stretch our "numeric" paradigm DP systems to incorporate new types of "information" or should we break away and "start over"?