

Editorial – Databases

The engineering principles established in the database field are so entrenched now that it is hardly surprising to find that the nine papers in this Database theme issue have a mathematical bias.

In the first paper 'Topological Approach for Testing Equivalence in Heterogeneous Relational Databases' we find that selecting on topological equivalence, before testing for dependency equivalence, is the quickest way of determining equivalence in practical database environments. This is because the number of primary keys is much smaller than the number of attributes in most database applications.

In real world databases we are beginning to see that data should never be deleted but should be archived when no longer of immediate use. With optical storage this idea has now become practical even at the desktop level. However auditors and analysts may need to refer to archived, historical databases and therefore the second paper 'Algebra and Query Language for a Historical Data Model' is topical. Sarda presents a 'state' oriented view of historical databases. Thus a temporal relation is a set of states where a tuple represents the state over a period of time; from which the state at a particular instance of time can be determined.

The problem of null values in databases is of particular importance now that the application of the relational model to office automation, text retrieval and image processing has been accepted through the use of nested relational databases. In the third paper 'The Nested Relation Type Model: An Application of Domain Theory to Databases' Levine and Loizou include inheritance in their model. Their approach provides rich data types for the analyst and lets the user define null types in addition to system defined null types. This enables incomplete information to be modelled in a very flexible manner.

Access to databases through graphical interfaces is becoming possible through the increasing use of workstations with bit mapped graphics. For example, CASE tools depend heavily upon graphics to represent the structure of systems and programs. In the fourth paper 'BRMQ: A Database Interface Facility based on Graph Traversals and Extended Relationships on Groups of Entities' Azmoodeh shows how the Binary Relationship Model Query language (BRMQ) is designed to help the end-user solve a complex query through the use of graphics.

As the volume of data stored each year increases rapidly we need more efficient ways of accessing large amounts of data. One way of achieving this is using a logical inference unit coupled to a relational database. In the fifth paper 'Integrated Deductive Database System Implementation' Bell *et al* show how logic

programming has been combined with the relational model to form deductive systems. A comparison is made at the logical, function and physical levels among 15 different approaches that have been documented over the past 6 years.

This idea of deductive databases is continued in Taylor's paper 'Logical Optimisation of Distributed Knowledge Base Queries' which addresses the problem of accessing many heterogeneous databases, each with its own data model. To enable access there needs to be a common knowledge model and a means of translation, which is achieved through the power of a frame language and Prolog. The query language is specified using the Vienna Development Method (VDM) and thus provides a useful introduction to formal methods. The language so defined has an optimisation technique which is a semantic preserving transformation.

The need to access distributed databases is continued in the paper 'An Evaluation of Site Selection Algorithms for Distributed Query Processing'. Martin *et al* comment on the opportunity of reducing the cost of queries in a large replicated and distributed database. Five techniques are explored: branch-and-bound, greedy, local search and simulated annealing.

The issue of security is highly topical and the paper 'Integrating Security with Fault-tolerant Distributed Databases' presents a protocol for tackling both security and reliability in a distributed database. This is important because fault-tolerant databases add to the security problem by replicating the same data object in many parts of a network. Agrawal and El Abbadi propose achieving security and resilience by integrating the information dispersal algorithm with a quorum protocol, using an integrated propagation technique. While this increases the overhead by maintaining several copies of the log in the system it improves security and resilience.

The distributed database theme is continued in the last paper 'Efficient Schemes to Evaluate Transaction Performance in Distributed Database Systems' which also deals with replicated data in a network. Estimates are made on a transaction's execution time, the reliability of the transaction, probability that access to another node will be required and the average number of data items accessed locally. These are vital considerations in a large financial network, for example.

All of the nine papers address problems of practical import to the database practitioner. A good starting point for such a reader might be 'Integrating Security with Fault-tolerant Distributed Databases' followed by 'Logical Optimisation of Distributed Knowledge Base Queries' (with its illustration of the use of Vienna Development Method) as these papers introduce issues that we must face in the 1990's.