Serdica

Bulgariacae mathematicae publicationes

Сердика

Българско математическо списание

NEW TRENDS IN DATA BASE MANAGEMENT SYSTEMS*

STEFANO SPACCAPIETRA

The analysis of Data Base Management Systems (DBMS) has recently climbed a major step with the ANSI/SPARC proposal, whose great merit was to give a world-wide publicity to the ideas of a multilevel DBMS architecture and the associated schemas. We first discuss the major concepts developed in that proposal and give a critical evaluation of their interest as well as of their limits.

On the other hand, the development of networks and network systems has significantly improved the amount of research being spent on the topic of distributed data bases, the very new trend in DBMS. We analyze here the very basic elements to be considered for the design of a distributed DBMS.

1. Introduction. Some new developments during the last two years have substantially changed the ongoing debates in the field of data base management systems (DBMS). Two of them are of primary importance: the introduction of the c o n c e p t u a l s c h e m a concept in a multilevel DBMS architecture and the current progress in the area of distributed data bases.

Altough these factors are not going to affect the present generation of DBMSs as actually under implementation by almost every computer manufacturer as well as many software developers, they will probably determine new guidelines for the next DBMS generation.

The purpose of this paper is to analyze the basic features supported by the conceptual and distributed approaches. The goal is to give an insight into the expected trends in the field of DBMS.

2. The ANSI/SPARC report. The April 71 CODASYL DBTG Report [1] has been an important milestone in the development of DBMS_1. It has been presented and discussed all over the computer world, until recently [2]; several CODASY-like DBMSs are currently being implemented and for some time the report has been considered as a possible candidate for standardization. The very idea of that report was to separate the data base description (called s c h e m a) from the multiple descriptions pertinent to a particular usage of a data base (called s u b s c h e m a s).

A similar idea is the primary basis of the ANSI/SPARC report, issued in February 1975. This report [3] generalizes the two-level CODASYL proposal into a multilevel system architecture (identifying 41 internal or external interfaces) and focuses on a three-level approach; its concern is on requirements, not on implementation problems.

2.1. T h e t h r e e-l e v e l a p p r o a c h. The report introduces three realms of interest in enterprise's data processing activity:

* Delivered at the Conference on Systems for Information Servicing of Professionally Linked Computer Users, May 23—29 1977, Varna.

SERDICA Bulgaricae mathematicae publicationes. Vol. 4, 1978, p. 219—223.

— conceptual: "including the limited model of the real world maintained for all applications";

— external: "including a simplified model of the real world as seen by one or more applications";

— internal: "including the data in computer storage representing the limited model of the real world".

At each level, a schema describes the corresponding data model. For each schema a person (more precisely, a role) in the enterprise is responsible for defining it and maintaining it: the enterprise administrator (conceptual schema), the application administrators (external schemas), the data base administrator (internal schema).

The advantages of introducing a conceptual level and schema are listed:

1) organazional: better (more stable, formalized) knowledge of the enterprise information model;

2) documentation: helpful to construct a data dictionary/directory;

3) control: helpful to ensure data security and integrity in a centralized approach;

4) simplicity: less mappings to be developed, if there is more than one internal schema;

5) data independence: insulate external schemas from changes in the internal schema and vice-versa.

2.2. Concepts for the Models. The report develops a data model for describing the real world and the schemas. Some generic concepts pertain to all models, some others are specific to a given model, especially for the internal schema, where additional concepts are needed to deal with storage structure.

The real world is described in terms of entities (whose scope is arbitrary), properties of entities, facts giving values to properties, entity-sets collecting entities together, and enterprise for the whole.

Description of the schemas relies on five generic concepts: field (smallest named object), group and record (both collection of fields and/or groups), plex and record-set (both collection of records and/or plexes). Relationships are implicitly included in the models, as they are utilized to construct groups. records, plexes and record-sets.

2.3. General Description of the System Model. A brief description of how schemas and programs are developed is useful to clearly understand how the proposed model works:

a) preparation of the conceptual schema. Four steps are identified in this process:

— identify and describe information needs in the enterprise (what information, how it is to be used);

— synthetize the information system: select informations to be managed, define security, integrity and avalaibility constraints;

— prepare a schema using a descriptive language for checking and documentation purposes;

— enter the schema into the DBMS.

b) preparation of the internal schema:

— first, once the conceptual schema is defined, the data base administrator has to analyze: usage requirements (as permitted) for data, system performance, timeliness of response, concurrency, security and integrity rules, implementation issues (access methods,. . .);

— prepare the internal schema:

— enter the schema into the DBMS.

c) preparation of the external schemas. The application administrators are responsible for determining their information needs, preparing the external schema and entering it into the DBMS. They have to transmit the schema to the application programmers and assist them in using the schema.

d) mappings. Mappings have to be defined between the conceptual and internal schemas (by the data base administrator) and between the conceptual and external schemas (by the application administrators).

e) program development. Once the permitted schema has been communicated to the programmer, he is able to write his program. At execution time, requests for data are transformed through successive levels of schemas, from the external level to a device/media access. A particular implementation could not require these separate steps.

2.4. Critical analysis of the report. We do not analyze here the other chapters of the report: they deal with detailed description of interfaces (the largest section in the report and helpful for standardization purposes), security, integrity, recovery and miscellaneous topics (these four chapters are merely an introduction to the dealed topic).

From a critical point of view, the great merit of this report was to give a world-wide publicity to the idea of a multilevel DBMS architecture and to emphasize the concept of a conceptual schema. Although these ideas were not unknown [4] the report has made them familiar to almost every DBMS specialst.

After two years of debates on the conceptual schema [5], [6], it appears that the usefulness of the concept is recognized, while a clear and accepted definition of what should be meant by these terms is still needed. The role of the conceptual schema as well during the enterprise analysis stage as during the DBMS operation stage should be more precisely defined. The content of the schema should also be discussed in more detail: integrity constraints, evolution rules [7], sharing specifications [8], mapping informations. Finally, it is not clear whether the external schemas are limited to a subset of the conceptual schema or not.

The most criticized part of the report is the section on the concepts for description of the data models. The proposed models had no supporters outside the originating study group; maybe that was a factor for the group to decide to abandon that model and probably turn over the most recent proposal in this domain, the entity-relationship model [9, 10]. In conclusion, the ANSI/SPARC report makes the CODASYL report obsolete for researchers and, in this sense, it may be regarded as the following milestone in DBMS development. The second planned version of the report is expected to cope with the deficiencies of the first proposal and will certainly originate a new wave of debates.

3. Distributed data bases. Networks and network systems are now operational in several countries. On the other hand, the state of the art in DBMS analysis bas been significantly improved by the three-level approach. The conjunction of these two factors of evolution in computer science is extremely propitious to the development of deeper investigations in the area of Distributed Data Bases (DDB). In most countries a large amount of research is being spent in this field [11], [12]; in France, a national research project started in 1976. It is worth examining the very basic elements to be considered for the design of a Distributed Data Base Management System (DDBMS)

3.1. Definiton and basic requirements. We call distributed data base a DB whose data are stored over possibly different computing facilities interconnected by a network system, with an integrated access to the data provided for multiple users interacting from several points in the network. In order to simplify the discussion, we suppose that data are locally stored into a DB using a local existing DBMS.

As for the DDBMS, it must be capable to avoid a to-and-fro movement between network nodes whenever an user's request calls only for locally stored data. This implies that the DDBMS itself is distributed over the network.

By these definitions we want to emphasize that we are not concerned with distributed file systems, DBMSs providing users with the possibility of simultaneously accessing several separate DBs, centralized DBMS with distributed data.

The development of a DDBMS, at a reasonable cost, should take into account three basic requirements.

a) The DDBMS should make use of the local DBMSs, which will perform all research, update and management functions on local data instead of and under control of the DDBMS. The expected benefit is to avoid reprogramming as well of system functions as of user programs. Of course, most of the problems arise when functionally different DBMS are used over the network: in this case, it is necessary to develop a common data model to describe the DDB and a common manipulation lanquate for communication between the DDBMS and the different DBMSs. Note that users go on using the same languages as before implementation of the DDBMS.

b) The DDBMS has to know as soon and as precisely as possible the location of the distributed data. This allows for efficient processing of users' requests. Wether the pre-existing data distribution over the DBs may be changed or not during the design of the DDB depends on the context in which the new system is developed: either to permit cooperation among different independent organizations, or to integrate several DBs existing in different branches of the same enterprise. The location catalog should be associated to the schema describing the DDB. A distribution unit has to be chosen: wether type or occurrence and to what extent (in a relational model: domain or relation).

c) Local usage of data should be permitted (using the local DBMS). The advantage is quite obvious: no change for programs not using data elsewhere stored. Unfortunately, this arises a difficult sharing problem, as the same data may be accessed through different overlapping schemas, one in the local DBMS and one in the DDBMS. Let us just mention that the problem can be solved, either by implementing a dialogue to ensure that integrity constraints are correctly checked, or by preventing the problem, splitting the DB into several disjoint parts, each one accessible through only one schema.

3.2. Different approaches. Different approaches are suitable for the design of a DDBMS. The first choice to be considered is on centralized versus decentralized control [13]. In a centralized approach, a unique description of the entire DDB is implemented. From this schema (corresponding to the conceptual schema in the ANSI/SPARC report) any other view of the DDB is derived. This allows for a unique control authority (the DDB administrator), which may be desirable for organizational, political or technical reasons. Note that centralized control does not mean that all DDB manipulations have to go through a central node to be processed.

On the other hand, different reasons (including psychological ones) may lead to the decision to implement the system so that each partner may keep his ownership, which implies control, of the set of data he allows to be integrated into one or even more DDBs. This approach is of course more complicated from a technical point of view, as the DDBMS has to cope with a plurality of overlapping descriptions of parts of the same DDB. A second choice has to be made in the design of the system architecture. Indeed, two possibilities exist; we call them local-oriented or distributed oriented architecture. In the latter the system is easier to implement but local requests are unnecessary time consuming; in the former the inverse is true (see [14] for a detailed discussion). Consequently, it is desirable to know, before system development, which of the local or distributed traffic is expected to be more important.

A decision on which of the possible approaches has to be followed and on which solution is to be implemented for the problem arised by local usage of data, is the very first and necessary step in the process of developing a DDBMS.

4. Conclusion. Multilevel DBMS analysis and distributed data bases are the up-to-date topics for investigations in the data base field. Significant improvements in DBMS development are to be expected from a successful issue of the many current researches.

REFERENCES

1. April 71 CODASYL DBTG Report — ACM, New York.
2. Data base description. Amsterdam, 1975.
3. ANSI/X3/SPARC study group on DBMS interim report, Feb. 75. ISO/TC97/SC5.
4. International workshop on data structure models for information systems — "Rapport introductif" — Namur. Belgium, 1974.
5. Modelling in data base management systems. Amsterdam, 1976.
6. Modelling in data base management systems, IFIP TC-2 Working Conf., Nice, France, January 1977.
7. E. Benci, F. Bodart, H Bogaert, A. Cabanes. Concepts for the design of a conceptual schema. *Modelling in data base management systems*. Amsterdam, 1976.
8. G. Gardarin, S. Spaccapietra. Integrity of data bases; A general lockout algorithm with deadlock avoidance. *Modelling in data base management systems*. Amsterdam, 1976.
9. P. Chen. The entity-relationship model. Toward a unified view of data. *ACM TODS*, 1, 1976, No. 1.
10. P. Moulin et al. Conceptual model as a data base design tool. *Modelling in data base management system*. Amsterdam, 1976.
11. J. P. Fry, M. E. Deppe. Distributed data bases: A summary of research. *Computer Networks*, 1, 1976, No. 2.
12. Journées AFCET bases de données réparties, Paris, Mars 1977.
13. P. G. Comba. Needed: Distributed control. VLDB 1975, ACM.
14. G. Gardarin, M. Jouve, C. Parent, S. Spaccapietra. Analyse de la conception d'un SGBD réparti. *Journées AFCET bases de données réparties*, Paris, Mars 1977.

Institut de Programmation
4, Place Jussieu 75005 Paris

Received 11. 9. 1977