Panel Section

Database Design Methodologies, Tools, and Environments

C. Batini (') and S. Ceri (")

(') Dipartimento di Informatica e Sistemistica, Universita' di Roma "La Sapienza"
Via Buonarroti, 12 - 00184 Roma
(") Dipartimento di Elettronica, Politecnico di Milano
Piazza Leonardo Da Vinci, 32 - 20133 Milano

List of panelists

Al Hershey, Database Design Inc., 2020 Hogback Road, Ann Arbor, MI 48104

George Gardarin, Universite' de Paris VI, 4 place Jussieu, 75230 Paris-Cedex 05

David Reiner, Computer Corporation of America, Four Cambridge Center, Cambridge, MA 02142

Presentation of the panel

Designing database applications is becoming more and more important, statistics indicate that over 30% of applications that will be built in 1985 will use a database management system, and thus database design is becoming a common practice in software development. This presentation focuses on the current state of the art of methodologies, models, tools, and design environments which can assist the design process, and indicates some emerging research topics.

1 METHODOLOGIES

Modern methodologies for database design agree on the well affirmed decomposition of the database design process into four steps

1. Requirements analysis and specification
2. Conceptual design
3. Logical design
4. Physical database design

Two new steps are usually included in recent methodologies

5. Distribution design
6. Prototyping.

Methodologies do not agree on where to place these steps in the traditional process e.g the DATAID methodology [CERI 83] places prototyping at the end of step 2, while the DDEW approach [REIN 84b] places prototyping at the end of the overall process. The interpretation given to Prototyping is different in the two proposals and motivates the different placement in [CERI 83] prototyping is seen as a verification that the conceptual model reflects user's requirements, while in [REIN 84b] prototyping is done using a DBMS as target and allows testing the efficiency of the implementation.

Recent methodologies reveal similarities between concepts and methods of conceptual design and more traditional approaches of software engineering

1. There are similarities between usual refinement techniques of program design and concepts refinement in the incremental schema enrichment of conceptual design [BROD 84].
2. Techniques like functional decomposition, cohesion and coupling of modules in software engineering are meaningful also for several phases of data design, like view decomposition [EWUS 84] or external schema design.
3. Documentation of data design can be performed by means of 'pure' top down refinements, similarly to what happens for program documentation.
4. The trend towards fast prototyping is common both to software engineering and database design.

Several topics still need a deeper understanding

1. Defining the set of qualities to be achieved at each stage of the design, and formalize a set of methods for their measure. E g. what shall we mean by 'readability' of a conceptual schema? Is it possible to measure readability and give a set of rules to restructure a schema in order to optimize such quality?
2. Several methodologies have been produced and experimented nowadays on nontrivial cases we need an evaluation of achieved improvement with respect to previous approaches. Conferences such [OLLE 82] are first attempts to give an answer to such questions.
3. The diffused need for reverse engineering requires the ability of maintaining conceptual schemas Suggesting strategies for maintaining schemas with hundreds or thousands of entities is a non trivial research area.

2 TOOLS

In these years, several tools have been developed for helping the database designer. Tools provide

1. Documentation features, for collecting and representing design data and the development of design sessions Inputs and outputs include

© 1985 ACM 0-89791-160-1/85/005/0148 $00 75

traditional forms and reports or more advanced graphic displays and editors, examples are Data Designer [DATA 81], DBDA [HUBB 81] and CINCOD [BATI 85]. For instance, GINCOD incorporates an automatic schema layout facility.

2 Design features, for solving specific design problems. Typical problems recently addressed are

a. Semantic analysis of conceptual schemas to determine completeness, correctness, minimality.

b. Restructuring and integration of different conceptual schemas.

c. Normalization of relational schemas.

d. Index and access path selection algorithms.

e Performance predictors or simulators for driving physical design.

3 ENVIRONMENTS

The recent trend in database design is to develop environments, i.e. integrated collections of tools. A non exhaustive list of environments includes Datail [CERI 83, ALBA 85a], DDEW [REIN 84b], Designer's Workbench [TROP 82], Galileo [ALBA 85a], IRMA [CURT 84], Taxis [MYLO 84, NIXO 84]. Environments are built according to two different coordinates

1. Approaches like DDEW aim to address all the phases of data design, providing an interactive support for specifying and experimenting data descriptions at the conceptual, logical, and physical level. The unifying feature of DDEW is the use of a same user's interface with menus, on-line help, windows, and graphic editing capabilities.

2. Approaches like Taxis and Galileo aim to integrate data design with software design, providing interfaces at different development levels, e.g requirement specification, functional analysis, program development and testing For instance, Taxis provides an object-oriented editor, a semantic consistency verifier, an interpreter, a debugger, a compiler into Pascal-P, an on-line help facility, and a documentation generator.

4 NEW APPROACHES TO DATABASE DESIGN

New areas for database design concern the emerging technologies, like distributed and heterogeneous systems, expert systems, and the new types of database applications.

Distribution design

In the last few years, we have noticed that many vendors of centralized DBMS have developed additional components for supporting the communication and cooperation between several instances of the DBMSs installed at different sites of the computer network [CERI 84]. As DDBMS enter the commercial arena, database designers will be faced by the following problems

a. Designing the distribution of a database schema (including objects and associations between them) This design problem can be decomposed into two subproblems

1 Determining the partitioning of data objects into subsets, called fragments, such that each fragment has the same properties with respect to data distribution.

2. Determining the allocation of fragments, possibly with replication, to the different sites of the DDBMS.

b. Designing the distribution of applications, in order to improve the performance of the system (costs and delays).

Notice that problem (b) is typically solved for a given data distribution, however in order to determine data distribution it is required to have at least an approximate knowledge of application distribution In fact, data distribution should be reconsidered during the life cycle of the distributed database, thus, design tools should be integrated with restructuring tools.

Interconnection of heterogeneous systems

Distribution design is useful for developing a distributed application from scratch, however, in many situations there is a need of interconnecting existing databases, possibly heterogeneous A solution to the problem, discussed in [DAYA 84], is to support the heterogeneous databases with a logically integrated global view, without requiring that the databases be physically integrated. The problems which are facing in designing the global view are the classical problems of view integration, however with different mechanisms for solving conflicts. In view integration, the designer has a major freedom in conflict resolution, while in database integration the designer is bound to existing data representations.

Use of expert systems for building design tools

A promising approach to database design (and, more in general, to software development) is the use of expert system technology for building design tools. The expert system should contain rules about the design process and facts about the system under design, the inferential mechanism should use deductive capabilities for determining which rules and fact can be used for inferring the structure of databases, at the conceptual, logical and physical level. For instance, [BOUZ 84] describes an expert system which acquires database definitions in the form of elementary sentences, uses them for building a semantic network, and then uses simplification, dependency, and normalization rules for generating first normal form relations and for normalizing them.

Rules introduced in the system can be independent from the specific domain of application (e.g , rules on the database design process), or else they can be domain-specific thus, specialized tools can be obtained for different domains of applications (e g banking systems, reservation systems, and so on).

New types of applications

New types of applications are now emerging statistical, CAD-CAM, office database. Such applications ask for new models and methods tailored to their specific environments. For instance, in statistical data bases, both elementary and aggregated data must be modelled. As a consequence, specific models for statistical database design are currently proposed [SU 83]. Experiences in statistical database design have convinced the authors that elementary data should be designed first, to guide an effective choice of desired aggregations.

REFERENCES

[ALBA 85a] A. Albano, L. Cardelli, and R. Orsini, "Galileo: a Strongly Typed, Interactive Conceptual Language", to appear on ACM - Transactions on Database Systems, 1985.

[ALBA 85b] A. Albano, V. De Antonellis, A. Di Leva eds., Computer-Aided Database Design, North Holland, 1985.

[BATI 85] C. Batini, E. Narlelli, M. Talamo, R. Tamassia, "GINCOD: A Graphical Tool for Conceptual Design of Database Applications", in [ALBA 85].

[BOUG 84] M. Bouzeghoub and G. Gardarin, "Design of Expert System for Database Design", in New Applications of Databases, G. Gardarin and E. Gelembe eds., Academic Press, London, 1984.

[BROD 84] M. L. Brodie, D. Ridjanovic, "On the Design and Specification of Database Transactions", in On Conceptual Modelling, M. L. Brodie, J. Mylopoulos and J. Schmidt eds., Springer-Verlag, 1984.

[CERI 83] S. Ceri ed., "Methodology and Tools for Database Design", North Holland, 1983.

[CERI 84] S. Ceri, G. Pelagatti, Distributed Databases Principles and Systems", McGraw-Hill, New York, 1984.

[CURT 84] R. M. Curtice, "IRMA: An Automated Logical Database Design and Structured Analysis Tool", in [Rein 84a].

[DATA 81] Data Designer Product Description, Database Design Incorporated, Ann Arbor, MI 48104.

[DAYA 84] U. Dayal, H.Y. Hwang, "View Definition and Generalization for Database Integration in a Multidatabase System", IEEE - Transactions on Software Engineering, vol. SE-10 no. 6, November 1984.

[EWUS 1984] F. Ewusi-Mensah, "Identifying Subsystems in Information System Analysis", Information Systems, vol. 9 no. 2, 1984.

[HUBB 81] G. Hubbard, "Computer Assisted Database Design", Van Nostrand, 1981.

[MYLO 84] J. Mylopoulos, A. Borgida, S. Greenspan, H. K. T. Wong, "Information System Design at the Conceptual Level: The Taxis Project", in [REIN 84a]

[NIXO 84] B. Nixon ed., Taxis '84: Selected Papers, Tech. Rep. CSRC-160, University of Toronto.

[OLLE 82] T. W. Olle, H. G. Sol, A. A. Verrijn-Stuart, "Information Systems Design Methodologies: a Comparative Review", North Holland, 1982.

[REIN 84a] D. Reiner ed., Special Issue on Database Design Aids, Methods, and Environments", IEEE Database Engineering, vol. 7 no. 4, Dec. 1984.

[REIN 84b] D. Reiner et al. "The Database Design and Evaluation Workbench (DDEW) Project at CCA", in [REIN 84a]

[SU 83] S. Su, "SAM: a Semantic Association Model for Corporate and Scientific-Statistical Databases", in Information Siences, 29, 1983.

[TEOR 82] T. J. Teorey et al., "Systems and Techniques for Research in Physical Database Design at the University of Michigan", in IEEE Database Engineering, vol. 5 no. 1, March 1982.