Lecture Notes in Computer Science

Edited by G. Goos and J. Hartmanis

402

Tapan P. Bagchi
Vinay K. Chaudhri

Interactive Relational Database Design

A Logic Programming Implementation

Springer-Verlag
Berlin Heidelberg New York London Paris Tokyo Hong Kong

Authors
Tapan P. Bagchi
Industrial and Management Engineering Programme
Indian Institute of Technology
P.O.-IIT, Kanpur 208016, UP, India

Vinay K. Chaudhri
TATA Consultancy Services
Nariman Point
Bombay 400001, MS, India

CR Subject Classification (1987): H.1.0, H.1.2, H.2.1, I.2.3, J.1

ISBN 3-540-51881-9 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-51881-9 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
2145/3140-543210 – Printed on acid-free paper

PREFACE

Rapid advances in data processing (DP) began about 1960 facilitating storage of data on magnetic records and files. This marked a major breakthrough in handling and organizing large volumes of accounting, sales, production, personnel or technical data, and gave DP a key foothold in the users' domain. However, these initial file-based data repositories had a "batch" orientation. The file-based approach focused primarily on the processing of transactions, providing only limited support to information management queries. No on-line queries were possible; the user could only get certain reports printed.

In the mid 70's the orientation of electronic DP started shifting. Users insisted that cost of data storage and maintenance be reduced by minimizing **duplication** of data, a serious shortcoming of these early databases. Interest also became strong in getting a database to store besides raw facts the **relationships** among real world objects. This would get closer to how humans discern real objects. Rarely are objects found to exist in isolation. They often relate to other objects in a variety of ways and the appropriation of this knowledge is clearly of considerable value. This shift paved the way for research on transforming stored data into **stored information**, to result in a multifold increase in the utility of a database.

Among the methods devised, the landmark work of Codd (1970) on the "relational model" representation of data quickly came to be regarded as a natural and efficient way of organizing information into databases. Codd's approach was conceptually simple yet formally elegant for it exploited the rich body of set theoretic operations to manipulate data. The key achievement of the relational model was that it showed how one could eliminate duplicate data and thus bring efficiency and integrity into storing information. Not surprisingly therefore many "relational" DBMS products soon became commercially available, promising to aid in the building of high capability databases.

A serious limitation nevertheless remains in these DBMS products. These products assume that the user already has available a good "relational structure" or a good **logical design** for his database, a fundamental requirement to gain advantages of the relational model. Getting the right relations for a database, however, is not yet a trivial step for the uninitiated.

The present work is a comprehensive logic programming implementation of the relational design methodology. It employs TURBO Prolog to test and establish computational viability of the relevant algorithms. It also presents the expert system prototype of a user interface, designed especially for builders of computerized databases who may have no formal training in relational (or any) database design.

The core of this interface is an expert dialog that exploits semantic information that a user can often readily provide about the objects of interest. Thus, rather than lean on the user's familiarity with database design formalisms, the approach discussed here engages the user in an emulated interview--as if the user is talking to a database expert. Internally the system combines Entity-Relationship modeling and normalization algorithms to automatically develop a relational structure which would satisfy design constraints up to the fourth or fifth normal form, ready for transcription to a DBMS.

The authors selected Gottlob's algorithm [14] for the projection of functional dependencies, Loizou and Thanish's method [19] to test loss-less joins, Ullman's algorithm [28] to test dependency preservation, Yao's summary of Bernstein's algorithm [30] to develop the third normal form and Ullman's method [28] to produce the Boyce-Codd normal form. ER modeling-type semantics are employed to handle multivalued and project-join dependencies.

This work has made some deliberate departures from a strictly analytical or algorithmic approach to database design. For instance inter-entity dependence is inferred by making a note of the dependence between/among keys of entities rather than by exhaustively checking each of the attributes of these entities.

Similar departures will be found in the great emphasis laid here on user interrogation with appropriate memory aids, examples and context-specific questioning. Integrity and consistency in user input are established later by lossless join and minimal cover check algorithms. These have enabled the present work to establish that major efficiencies can be achieved in the time required to reach the final design when the user's general knowledge of objects and the environment in which the database will be used is folded as needed into the machine-assisted design process. The result is a significantly faster design process, especially when rigorous approaches would employ algorithms that may be NP-complete.

ACKNOWLEDGMENTS

The authors express their gratitude to the faculty of the IME Systems Optimization Laboratory and the Computer Science and Engineering Department, IIT Kanpur who helped with advice and countless sessions of invaluable discussions as this work progressed.

T.V. Prabhakar, R. Sangal and J.L. Batra provided constant inspiration and feedback. Santosh Gupta, Prashant Kumar, Ashok Mittal, S. Sadagopan and Kripa Shanker abundantly gave material and moral support--often by sacrificing their own priorities. The authors are indebted to the M. Tech. IME classes of 1986 and 1987 who acted as subjects in most of the prototyping work.

The authors also thank their gurus, Jim Templeton (University of Toronto), and H.C. Agarwal, M.P. Kapoor, M.M. Oberai, and A.S. Parasnis (IIT Kanpur) whose teaching and guidance made this long journey of research eventually possible.

This work is dedicated to Vinay Chaudhri's family and to S. Anupam Bagchi and Arun P. Bagchi.

CONTENTS