M. VETTER
R. N. MADDISON

DATABASE DESIGN METHODOLOGY

Database Design Methodology

M. Vetter
(IBM ESRI)

R.N. Maddison
(The Open University)

Prentice/Hall International

ENGLEWOOD CLIFFS, NEW JERSEY LONDON NEW DELHI
SINGAPORE SYDNEY TOKYO TORONTO WELLINGTON

British Library Cataloguing in Publication Data

Vetter, Max
Database design methodology.
1. Data base management
2. File organization (Computer science)
I. Title
001.6'442 QA76.9.D3

ISBN 0–13–196535–2

Library of Congress Cataloging in Publication Data

Vetter, Max, 1939–
Database design methodology.

Bibliography: p.
Includes index.
1. Data base management. I. Title.
QA76.9.D3V47 001.6'42 79–13876
ISBN 0–13–196535–2

ISBN 0–13–196535 2

PRENTICE-HALL INTERNATIONAL INC., *London*
PRENTICE-HALL OF AUSTRALIA PTY., LTD., *Sydney*
PRENTICE-HALL OF CANADA, LTD., *Toronto*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
PRENTICE-HALL OF SOUTHEAST ASIA PTE., LTD., *Singapore*
PRENTICE-HALL, INC., *Englewood Cliffs, New Jersey*
WHITEHALL BOOKS LIMITED, *Wellington, New Zealand*

Printed in the United States of America

81 82 83 84 85 5 4 3 2 1

Preface

The developments of database facilities and applications in recent years have created a need for, and made possible, a basically new approach to database design. Designers need methods of successfully analyzing the kinds of information that will flow to and from users and be represented as data in computerized systems. Their aim is efficient, cost-effective and logically right data models. They must implement, not merely to meet current and likely applications within available and anticipated technology, but also build in flexibility to meet any future evolution without expensive reprogramming or restructuring.

Database analysis and design is passing from the phase of research and a collection of techniques that have been shown as useful in practice. We discuss the various theories and techniques, bringing together the most promising into a single comprehensive systematic procedure for analysis and design, yet indicating that ours is not the only way.

By coordinating that which is currently available from many sources this book will assist in the improved use and expansion of existing DBMSs. More importantly it provides guidelines for the evolution of future DBMSs which may as yet be only reasearch ideas.

* The design method proposed in this book, and which we and others believe to be necessary, is therefore new. It differs from other approaches by developing an application–independent analysis of the information in an organization to produce a structure which is also software- and hardware-independent.

* The design procedure is based on proven theory and the combination of tools proposed is unique. It applies set theory, functions, the theory of graphs and normalization rigorously to the database design process. We show every principle by example and state the general.

We explain how to develop a conceptual data model which reflects the inherent properties of the information, independent of current applications and of technical limitations and will stand the tests of time and evolution. From this stable reference point are developed specifications of the interfaces and data structures to be used for all kinds of DBMSs so that future changes, large or small, external or internal, general or specific are as easy as possible. Different types of structures and interfaces can exist together.

Database terminology varies: by and large we have followed the American National Standards Institute Computers and Information Processing Standards together with conventions of school, college and university mathematics and computing.

The book is aimed at a wide readership:

* Data processing professionals such as data analysts, systems analysts, database designers, database administration staff and programmers seeking promotion may learn from the mixture of theory and practical examples.

* Students and teachers of university or similar courses in computer science should also appreciate the compilation of fundamentals which otherwise would have to be gathered from many books, journals, reports and conference proceedings.

The table of contents summarizes the structure of the book; this is expanded in the introductory Chapter 1 which outlines how the various topics are developed and how they fit together. There are questions and exercises at the end of each chapter, as well as references and bibliographies for further study. Solutions are in an appendix.

The material for this book comes from courses developed and taught at the IBM European Systems Research Institute (ESRI) in La Hulpe, Brussels, the AMAGI Research Institute in Itoh, Japan and the Swiss Institute of Technology in Lausanne; and also from various papers given at international conferences in Japan and Europe.

Comments and suggestions from colleagues and reviewers have been invaluable in shaping and refining the structure of our approach. In particular we are most grateful to Professor C.A. Zehnder (Swiss Institute of Technology), Professor P. Wilmes (University of Louvain, Belgium), members of IBM ESRI and to Dr. M.J. Beetham and the Open University Student Computing Service Research Computing Advisory Service.

Any errors or omissions are ours however, and any suggestions for the book's improvement would be welcomed.

M.V.

R.N.M.

Contents

1

Aims

1.1 INTRODUCTION

Operational and management information, when represented as data usable in many ways, is a valuable resource. Particularly it is so if it can be easily and quickly used to meet all possible requirements - answering the unexpected as well as the routine.

This means an organization should design the way that its data is held so that it can easily be used for any application and so new applications can be added without requiring any extra work on previous ongoing ones.

A database is a collection of data stored and organized so that all user requirements can be met.

By a user we mean an individual or group that has requirements involving providing or receiving information. This excludes application programmers and computer systems staff.

The structure, design and control of a database normally need a database administrator (DBA). This is a person or team that controls and manages the database. The technical responsibilities of the database administrator are to

- identify the needs of the organization and of users
- define, implement and control the data storage including the structure and self-consistency
- define and control access to the data
- coordinate the data resources of the whole organization, ensuring user and management cooperation.

The job needs tactful politics to ensure success. Policies and procedures have to be established to guarantee effective protection and to control use of the data. Coordination and agreement are also needed over the choices of the overall information areas to be covered.

Traditionally organizations are divided into departments, divisions, groups and sections with responsibilities for particular aspects. The functional areas, i.e. groups of related business system activities and processes, may or may not be the same as the departmental structure.

The information subject areas are those overall groups of kinds of information to be held and communicated by computer. These information areas will cover the information common to the functional areas whose staff use the computer. E.g. employee information is a possible information subject area; it will be used in functional areas such as payroll and personnel. Similarly sales information is an information area that is used in activities such as sales order processing and other activities for accounting and production planning.

The database administrator's responsibilities require actions that are far beyond the activities required of a data processing department implementing traditional file-based computer systems. Briefly we review such systems to show the difficulties.

Such systems were each tailored to meet specific operational application requirements. These requirements were identified and documented by systems analysts. Their corresponding application programs were each designed and coded almost independently. They each used files of records structured individually to meet the needs of the application.

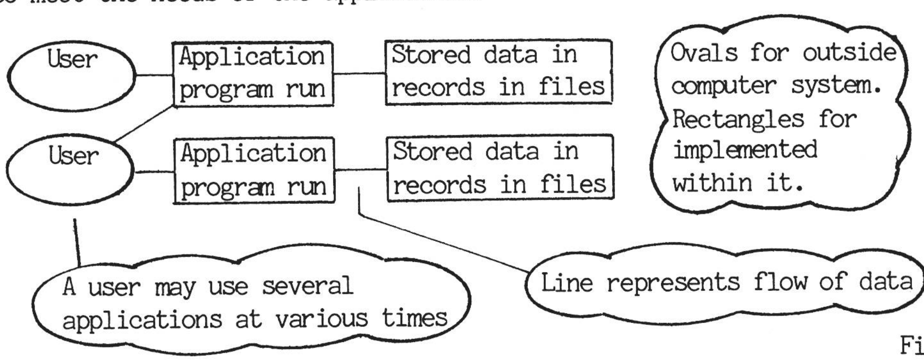

Figure 1.1.1

Little thought was given to holding the data in structures that could easily be used in new ways, e.g. in a new application that associated together data from several types of record in previously unrelated files.

Gradually more application programs were added. For effectiveness on the computers of the 1960s and 1970s some processes had to be batched, i.e. a group of similar transactions were processed together. Because one application process needed data in files sorted one way e.g. by customer account number and another by e.g. invoice number, many sorting and merging programs and processes were also needed. This involved many versions of the data in files on magnetic tapes. Such data could only be accessed sequentially. Computer runs had to be done in correct sequence and weekly or monthly. The availability of disk storage made possible record access in any sequence (called random access) by application programs. The applications were programmed in terms of logical records that were stored by the operating system as say five per physical block on disk.

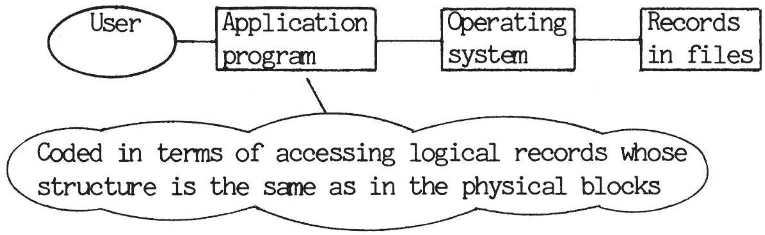

Figure 1.1.2

For the moment logical means as apparently structured, e.g. for handling by an application program.

But the data structures were still chosen to fit individual applications. The choices of what types of data items were in which record types and what pointers from one record to another existed were highly application dependent.

Database management software

The 1970s saw an extra software interface, a database management system (DBMS). This is between the application programs and the accesses to the physically stored data. So the application programs can be coded as though they view the data in seemingly different structures from the actual physical structures. There is no sorting, merging or sequential file processing.

Because of their enormity database management software packages are usually general purpose. They are developed by a computer manufacturer or software house. In fact they contain several interfaces for reasons described later.

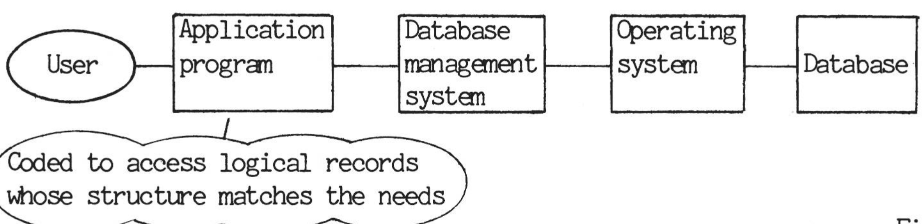

Figure 1.1.3

In this arrangement a physical block of data on disk storage may contain a mixture of data whose structure in detail is not deducible from the application program coding.

The above leads to these concepts, the first four being desirable features of a database management system

- all information within the subject areas relevant to users should be storable

- each application can be programmed as though only the kinds of data that it needs are present
- new applications and new types of data can be added without disturbing the coding or operation of existing applications
- only the database management system directly interfaces to the physically stored data, the application programs do not
- particular computer equipment and software, including database management systems, can only do what they have facilities for. The limitations may constrain the designer.

We shall gradually develop these ideas.

Subjects of interest

To the database designer and administrator the areas of interest about information and data are as follows
- the real world and its information, e.g. ABC is a manufacturing organization
- that information concerning the real world that is within the areas relevant to users, e.g. J. Smith orders 5 engines from ABC's salesman, but not J. Smith's height. We call this the mini world.
- ideas in people's minds about such information, e.g. there are orders concerning quantities of parts for customers. This gives a number of associated types of information called a conceptual data model for reasons below
- ideas about the flow and use of such types of information, e.g. the orders are processed in the Sales Order Processing section, giving a functional model
- the ways of representing all such information types as data meeting computer constraints, e.g. having certain named data items such as Order-date in a logical record type named Order
- the physical structure and representation of the actual data to be held on storage devices such as discs, possibly with transfers controlled by an operating system
- the logical structures of subsets used by applications
- the specification of application processes
- communication and transmission of data
- the control of resources, including the initiation and execution of processes by the computer
- security.

The conceptual data model describes the structure of all the types of data needed to represent the mini world information. A functional model describes the data flows and use for applications e.g. for purchasing.

Usually no user is involved with every type of information but the conceptual data model includes all types.

One of our central ideas is that developing the conceptual data model should be done independent of computer constraints. During this modelling the analyst should discover the structure of the information by talking to users and managers - never minding what can or cannot be done by the available database management system, computer and operating system. Only thus can users' requirements be documented satisfactorily. After finishing that conceptual modelling and getting users' and management agreement - and only then - the designer should start to think about the global logical model i.e. the way the data is logically structured in the computer system.

We hope you, the reader, are not put off by all the terms used. They will all be gradually explained in the next few chapters. For the moment

- logical means as apparently structured
- global means covering all types of information
- conceptual means as formed in the mind - free of any computer considerations.

To aid data modelling, the analyst and database administrator may need to use a computer system to keep records of the many types of data items involved. Some database management systems provide facilities for storing the names of the types of data items e.g. Employee-no, Order-date, and a description of each. This is called a data dictionary. A data dictionary can in principle hold the details of

- the global conceptual data model
- the global logical model
- each application program logical model
- access, integrity and resource controls
- the physical model.

The details should include how the data types are associated to carry meaning. The details should also include in which applications each type of data is used. Since the associations that are the essence of the meaning are stored, the data dictionary can be used to analyse what would be the effects of proposed changes in the data types and associations.

Only if a change occurs in the real mini world as modelled in the global conceptual data model should that model need to be changed. Thus only then should the global logical model change. Exceptionally the global logical model may also need changing if the current database management system that it fits is to be replaced by another, though strictly that is the development of a new global logical model for a different set of constraints. The usual case is a new software release with advantageous features. The coded form of a global logical model is called a schema. It is the way the data in the database is regarded as structured.

All computer systems need features for privacy and integrity - not just database systems. Security features cover access control - information should not be disclosed to or corrupted or destroyed by unauthorized people, e.g. some people may be able to create, modify, and delete an employee's data, others to read only, others only to do overall processes giving anonymous statistics.

All systems need accuracy - information should agree with reality. E.g. some data may have an 'owner' responsible for its accuracy and applications will have data vetting checks. Integrity means the stored information should be self-consistent and not get corrupted or lost by accident or malfunction. This needs recovery arrangements to rectify the stored data after hardware and software errors, power failures and other misfortunes.

Finally the system must be useful and easy to use. Information must be received by the correct people at the right time. They must understand the data they receive so they can act on the information. They must have confidence in its quality. They must be able easily to obtain information that they need and to feed to the computer system data representing new information. So the computer system must be reliable and easily usable to be useful.

Stages of design

The design and development of a computer-based information system has various stages. Most projects should include the following activities. Arrows denote the flow of information.

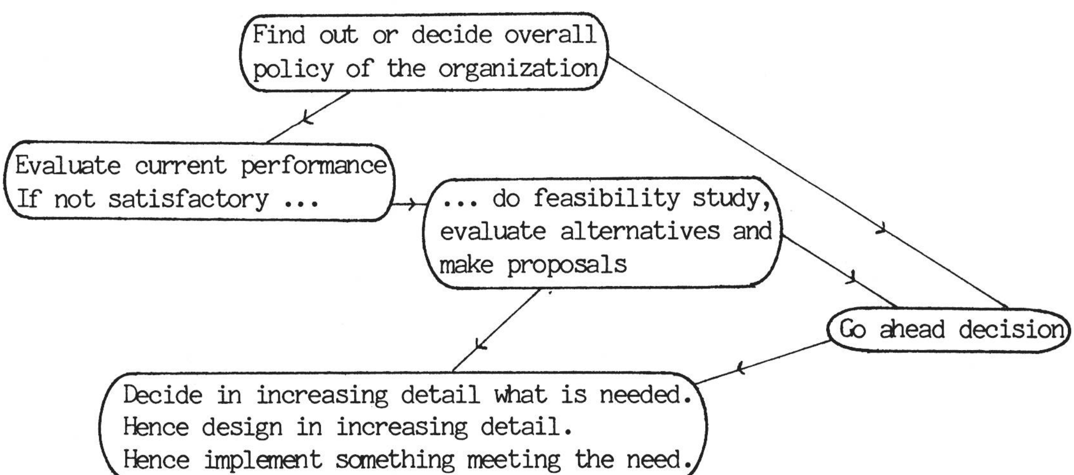

Figure 1.1.4

Analysis comes before design. For computer-based information systems the last box above includes something like the following.

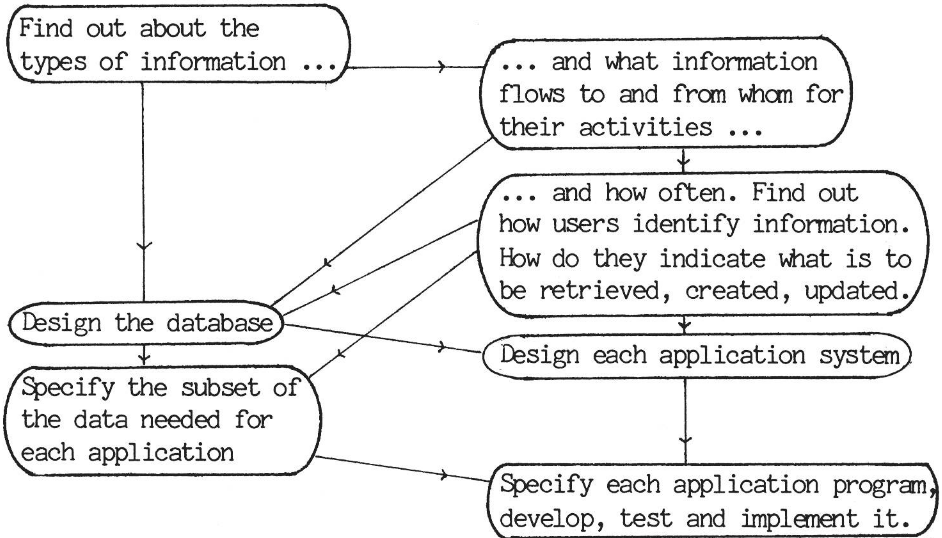

Figure 1.1.5

In the above the left side is mostly about types of information and types of data, i.e. data models. The right side is mostly about users' applications and systems, i.e. business functions and functional models.

Design aims

Before tackling the various stages and methods of analysis and design you need to appreciate the aim. The aim is to design a database

(a) to serve many applications

(b) so new types of data and new applications can be added easily

(c) so that as the pattern of use evolves the structure of the data as held on disk can be changed without affecting application programs

(d) so all types of information can be stored, even though no user knows about all the types of information

(e) that conforms to proposed or agreed standards

(f) that uses a particular software package, e.g. a database management system available on a particular computer.

These six considerations are always true.

There are also features of the organization that are likely to be true for ever. For example it employs staff and manufactures goods for sale.

The aim is to create a database design that embodies and is based on

- the consequences of the six ever-true considerations
- those statements about the organization's information that are true long-term.