Computer Science Journal of Moldova, vol.**1**, no.1(1), 1993

A system for conceptual and logical database design

V.Cotelea A.Caranicolov Ia.Ceban Iu.Ciubota

Abstract

The general scheme of a system for a conceptual and logical database design is described.

The database management systems, becoming a convenient way of creating a database, are frequently used as a new mode of access to data, but not as a means of common possession of them. This is due to the great importance of database structure. The principal consequences of an unsuitable structure are the unmeasurable expenses for its modification which must be realized as soon as the inconsistency of this structure is found.

Hence, one important factor in the exploration of database is the existence of a methodology and a software which can be used for automation of the database schema design .

The proposed system consists of two phases : conceptual design and logical design (see Figure).

In the conceptual design phase there are the following stages: the information requirements analysis, views conceptual design and views integration.

The information requirements analysis is the first stage of the database methodology. During this stage the designer collects the user's information requirements concerning those parts of the organization which are to be automatized by means of a database applications, and formalizes them into descriptions of data, operations and restrictions.

The information requirements analysis is related to the elaboration of:

- information requirements expressing data and their properties;
- operation requirements describing which operations have to be performed with the data;
- constraint requirements expressing restrictions on data and operations;

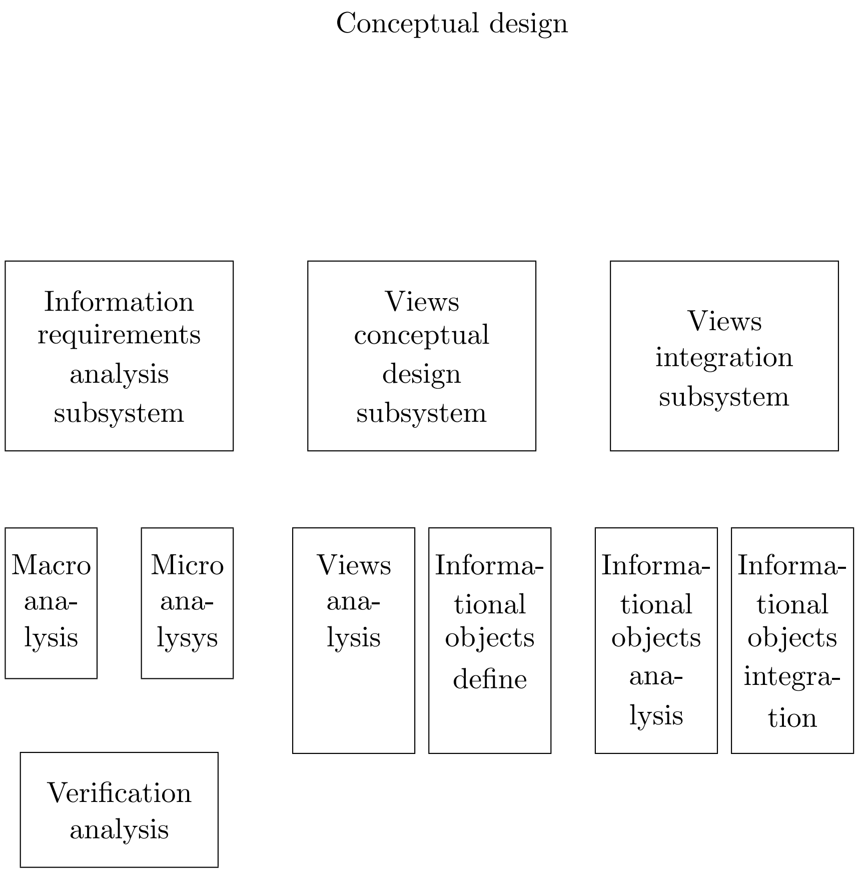
At the second stage of conceptual design, called views conceptual design, each user's conceptual views are defined independently. Conceptual views are formalized by representation for both statical requirements (data) and dynamical requirements (operations). Data modeling treats the semnifications of data, therefore the central issue is the choice of a suitable data model which allows one to express properly the application of semantics. Here the enriched Entity-Relationship model has been chosen.

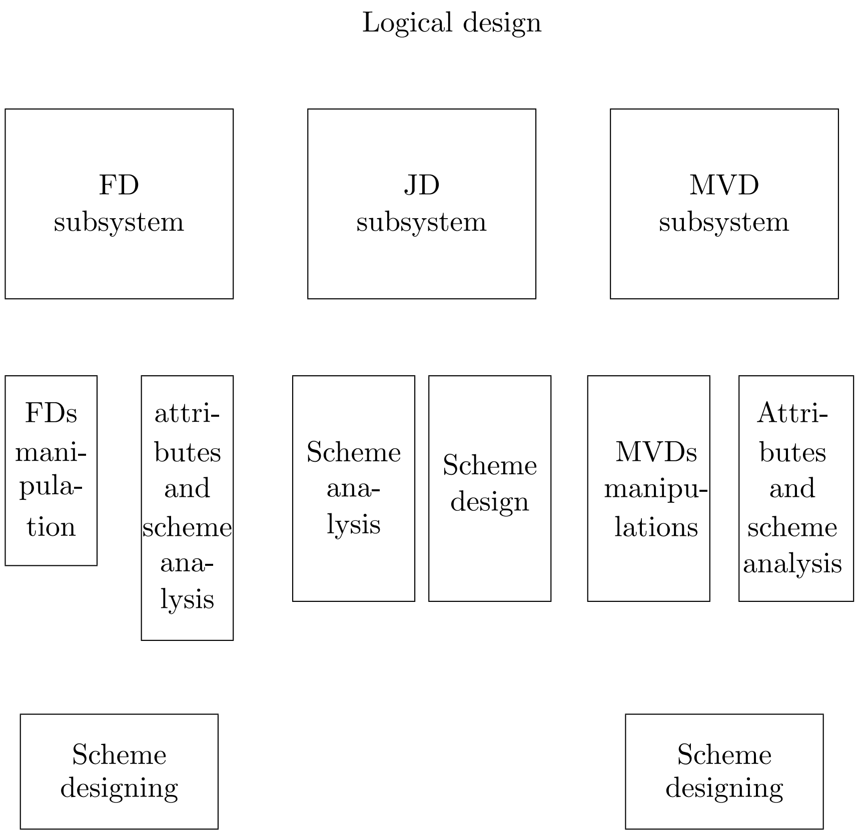
The design of conceptual views may be in general developed independently by different designers and at different time, because the users purposes are different and different are the equivalent interpretations in model.

The last process of conceptual database design is views integration. At this stage, the conceptual views, that have been defined independently during views conceptual design, are combined into a unique global view of the application. Several complex activities are required during views integration:

- finding the common parts between different schema;
- finding the different representations chosen by designers;
- discovering inappropriate or unreliable choices;
- determining inter-schema properties, i.e. a properties involving data belonging to different schema and which involuntarily "masked" by designers.

The product of conceptual design phase is the Global Conceptual Schema.


Conceptual design
Information requirements analysis subsystem
Views conceptual design subsystem
Views integration subsystem
Macro ana-lysis
Micro ana-lysys
Views ana-lysis
Informa-tional objects define
Informa-tional objects ana-lysis
Informa-tional objects integra-tion
Verification analysis


Logical design
FD subsystem
JD subsystem
MVD subsystem
FDs manipulation
attributes and scheme analysis
Scheme analysis
Scheme design
MVDs manipulations
Attributes and scheme analysis
Scheme designing
Scheme designing

The phase of conceptual database design is being elaborated currently and will be realized in TURBO PROLOG.

The phase of logical database design involves: the FD subsystem, the JD subsystem and the MVD subsystem.

The purpose of the FD subsystem is to automatize the database schema analysis and design, the integrity constraints of which are functional dependencies. The FD subsystem solves the following problems:

- finding the closure of attribute with a set of functional dependencies;
- membership problem, i.e. derivation of a functional dependence from a set of functional dependencies;
- determining if two sets of functional dependencies are equivalents;
- finding a nonredundant cover for a set of functional dependencies;
- partition of a set of functional dependencies into equivalence classes;
- finding a minimal cover for a set of functional dependencies;
- construction of a set of reduced functional dependencies from a given set of functional dependencies;
- sorting of a set of functional dependencies;
- designing of a database scheme in the third normal form;
- determining if a database scheme is in the second normal form;
- determining if a database scheme is in the Boyce-Codd normal form;
- finding a determinant;
- finding the set of candidate keys for a given relation scheme.

The problems mentioned above conventionally are divided into three groups: the dependencies manipulation (problems 1-8), the database schema design (problem 9), the database schema analysis (the last four problems).

The database scheme can be viewed as a hypergraph. Such a structure is described in the JD subsystem.

It is known that a large number of desirable properties of database schemes fall into a small number of equivalence classes, each completely characterized by the degree of acyclicity of the associate hypergraphs. For example: if the scheme is acyclic, then the database is globally consistent if and only if it is pairwise consistently; there is a semi-join program that fully reduces all of the relations in the database; a scheme is acyclic if and only if there is a monotone join expression, etc. Moreover, for acyclic schema there are efficient (polynomial-time) algorithms for solving problems that are NP-complete in the general case. What is more, the acyclic schema allows one to elaborate effective algorithms for optimizing the access to data.

For today the following problems from JD subsystem are solved:

- determining if a given database scheme is α-acyclic;
- determining if a given database scheme is β-acyclic;
- determining if a given database scheme is γ-acyclic;
- determining if a given database scheme is Berge-acyclic;
- determining the degree of acyclicity of a given database scheme.

It must be mentioned that there is not found a sound and complete set of inference rules for join dependencies. However a desired property here is that there exists, for acyclic database schema, an associated set of multivalued dependencies. And such a set of inference rules is found for multivalued dependencies. Therefore in the phase of logical design the MVD subsystem is included, which solves the problems of multivalued dependencies manipulation, of analyzing and designing the database schema the logical constraints of which are the multivalued dependencies.

The subsystems of the phase of logical design are realized in TURBO PASCAL.

V.Cotelea, A.Caranicolov, Iu.Ciubota
Institutul Naţional de Cercetări
Economice şi Financiare,
59, Bănulescu-Bodoni str., Chişinău,
277001, Moldova

Ia.Ceban,
Institute of Mathematics,
Academy of Sciences of Moldova
5, Academiei str., Chişinău,
277028, Moldova

Received September 3, 1992