INFORMATION PROCESSING 74 – NORTH-HOLLAND PUBLISHING COMPANY (1974)

RELATIONAL MODEL FOR A DATA BASE*

C. DEHENEFFE, H. HENNEBERT, W. PAULUS

Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix
Namur, Belgium

Starting from observation of real situations we decided that it was more natural to consider two types of relations : individuals defined by n-ary relations defined on sets of elementary values or characteristics, which represent items of information discovered in the outside world, and binary relations which represent semantic relations between individuals. In the rest of the article, we present a few considerations on a consultation language for data banks.

1. INTRODUCTION

The aim of our paper is to present briefly the essential ideas in the relational model and to outline the language which manipulates the entitites defined in it ; we will not discuss other topics such as the access model which, with the relational model , constitutes the complete architecture of our system.
In our view, every problem presupposes that we first isolate "entities" of information that appear naturally and then the semantic relations which connect them. Starting from this hypothesis we describe the relational model which is its formalism. We consider principally"INDIVIDUALS" as n-ary relations defined over a certain number of "CHARACTERISTICS" and the semantic relations as binary relations between individuals. We will try to relate our approach to the work of E.F. Codd [1],[2],who considers that a natural approach is to view a data base as a collection of tables having a certain number of characteristic properties and varying in time. He only considers first order relations and we can say that the concept of key which allows him in a certain way to represent second order relations between primary relations is not explicit,contrary to our approach.The language which he next defines,manipulates relations allowing him to achieve independence of the programs from the physical structure. However,without neglecting this important objective we considered it necessary to define a model which represents the reality as faithfully as possible. We are used to manipulating entities which we consider significant in the problem being treated, with these entities linked by relations. Starting from this observation we have isolated the concepts of individual and relation.

Once this has been realized and described by the relational model, it is obviously necessary to have a language capable of manipulating the defined entities simply. We think that only a conversational language will allow us to realize this objective of simple manipulation, since such a system can help the user to define his requests.

As for the problem raised above,of the independence of the user programs with respect to the data,which is an essential part of the design of a data bank system, we think that a hierarchy of models will solve this problem. Without going into further details, we envisage on the one hand a relational model as a tool for describing the semantics of a problem, and on the other hand,an access model defined in such a way as to describe and resolve the problems of efficient representation, search, protection and maintenance and a physical model describing the implementation.

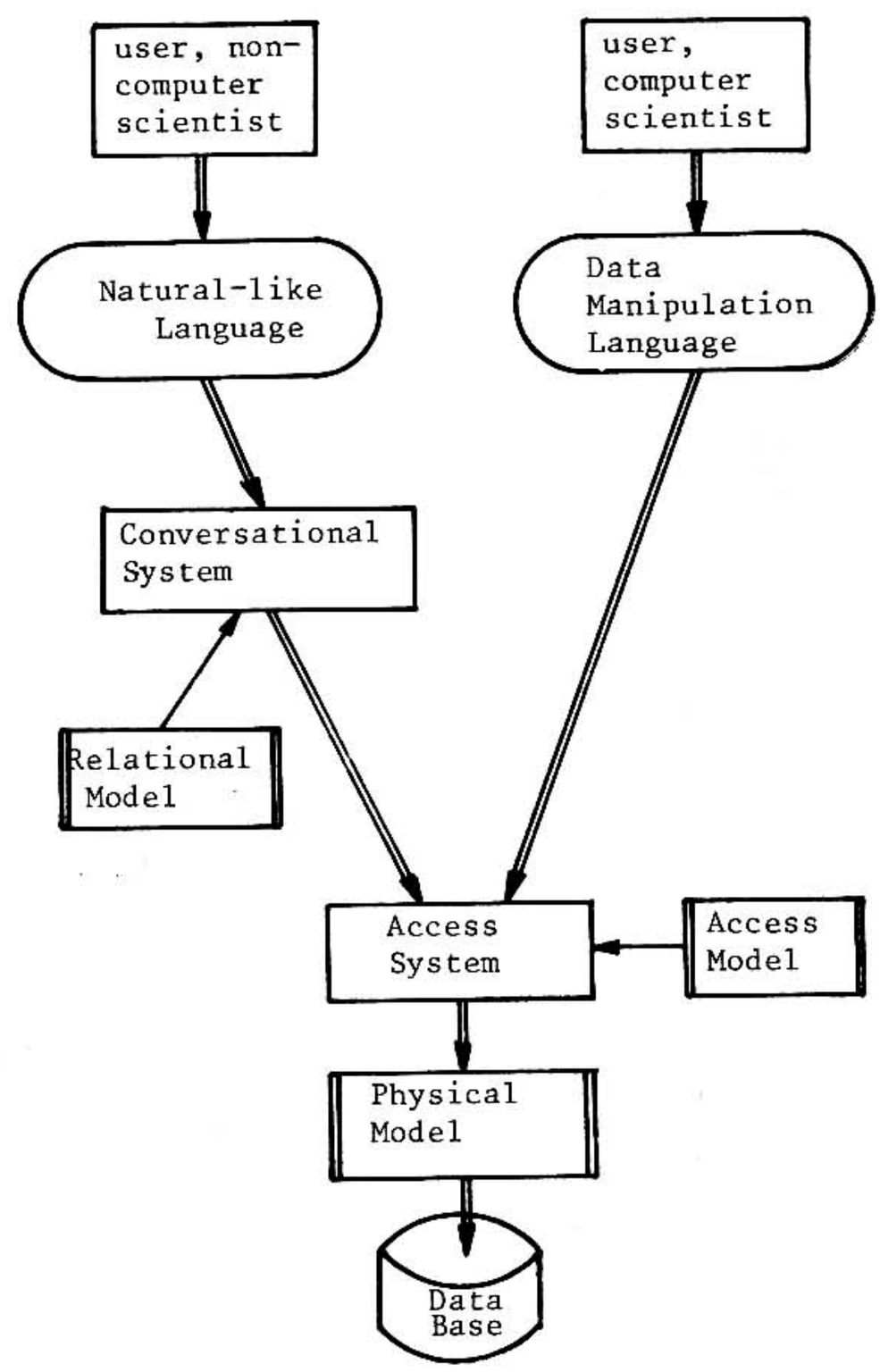


Fig. 1. Architecture of the system

2. DESCRIPTION OF THE RELATIONAL MODEL

2.1 Concepts

As we have already said, even a user who knows nothing of computer science, is used to manipulating items of information which he considers significant for his problem and these items are connected by relations;we have introduced the following concepts in order to take this hypothesis into account.

(i) The concept of characteristic

A characteristic is a pair (name ,V)where V is a set of atomic values(a sequence of alphanumeric charac-

* Within the framework of the national program for the advancement of research in computer science, sponsored by the Commission Interministérielle de la Politique Scientifique.

ters). A characteristic is thus designated by its name. For example, if we have the characteristic (AGE, {1,...,79}), we will say that 18 is a realization or value of the characteristic called AGE. The pair (AGE, 18) constitutes an item of information ; V is called the domain of the characteristic.

(ii) The concept of individual

An individual is a pair (name,R) where R is an n-ary relation defined over n characteristics, where the order of the domain within the relation is not important. For example, if we have three characteristics (NAME, {Dupont, Durand, Martin}), (SEX, {male, female}) and (AGE, {1,...,79}), we could define the individual (PERSON, R) where R is the set : {{(NAME, Dupont), (AGE, 17), (SEX, male)},{(NAME, Durand), (AGE, 19), (SEX, female)},{(NAME, Martin), (AGE, 27), (SEX, female)}} R is called the domain of the individual. In order to remember that the individual is defined over the characteristics NAME, AGE, and SEX, we will write PERSON (NAME, AGE, SEX).

(iii) The concept of relation

This concept allows us to represent a semantic relation between two individuals. A (semantic) relation is a pair (name, R) where R is a binary relationship defined over two individuals, the order of the domains of the relation having no importance : they are identifiable not by their position within the relation but by their name. This fact implies of course that the name of a domain is unique within a relation.

For example, if we have defined the individuals named PERSON and FIRM, we can define the relation denoted by EMPLOYER (PERSON, FIRM) as (EMPLOYER, R) where R is, for example :
R={{(PERSON, {(NAME, Dupont)}),(FIRM,{(NAME,"IBM")})},
{(PERSON, {(NAME, Durand)}),(FIRM, {(NAME, "ICL")}},
{(PERSON, {(NAME, Martin)}),(FIRM,{(NAME,"UNIDATA")})}
If the two members of a relation are identical, we must qualify them in order to distinguish them semantically. Example : if we define a relation as : (FILIATION, {{(PERSON, {(NAME, Dupont), (CHRISTIAN NAME, Jean)}), (PERSON, {(NAME, Dupont), (CHRISTIAN NAME, Nicolas)}),.....}.
We note that it is ambiguous. So it is necessary to define it as : (FILIATION, R) where R is {{(PERSON. PARENT, {(NAME, Dupont), (CHRISTIAN NAME, Jean)}), (PERSON.CHILD, {(NAME, Dupont), (CHRISTIAN NAME, Nicolas)})}..... {.....}}.
Thus, the user's job is to define on the one hand, named individuals, formed by grouping a certain number of named characteristics and on the other hand relations between these individuals. The definition of these individuals and relations depends heavily on the problem treated. An individual CLIENT, for example, is completely different if defined by the accounts department than if defined by the sales department.

(iv) The state of a data base

We will denote by state of the data base, a collection of individuals and relations, completely defined at time t. It is clear that the items of information evolve in time, more precisely, the individuals or relations can be modified (addition or suppression of elements of sets) and new individuals or relations can be defined or existing ones suppressed. Thus, we can state that a data base is a countable sequence of states.

2.2 Types of relation

A relation R (A, B) defined over the individuals A and B is :
1-1 if $\forall a \in A \quad \forall b \in B \quad |R_A[a]| \leq 1$ et $|R_B[b]| \leq 1$.
1-n if $\forall b \in B \quad |R_B[b]| \leq 1$.
n-1 if $\forall a \in A \quad |R_A[a]| \leq 1$.
m-n in the other cases.
We use the notation || for the cardinal.

This classification permits a realisation of important semantic properties of the relations, for example, if FATHER (PERSØN, PERSØN) is a relation defined for the individual person, it is 1-1 and in this way we express the fact that a person can have only one father. The importance of this is not negligible for the implementation, the m-n relations are not implemented in the majority of systems, as they are very difficult to do.

2.3 Existence properties of individuals

The realisations of an individual do not always exist independently of those of one or more other individuals : for example, an individual "ADDRESS" has realisations only if they are effectively the addresses of one or more people, an individual CHILD has elements only if they are the children of one or more people. An examination of such situations has led us to describe the following typology of individuals based on their property of existence.

(i) Obligatory member of an individual with respect to a relation

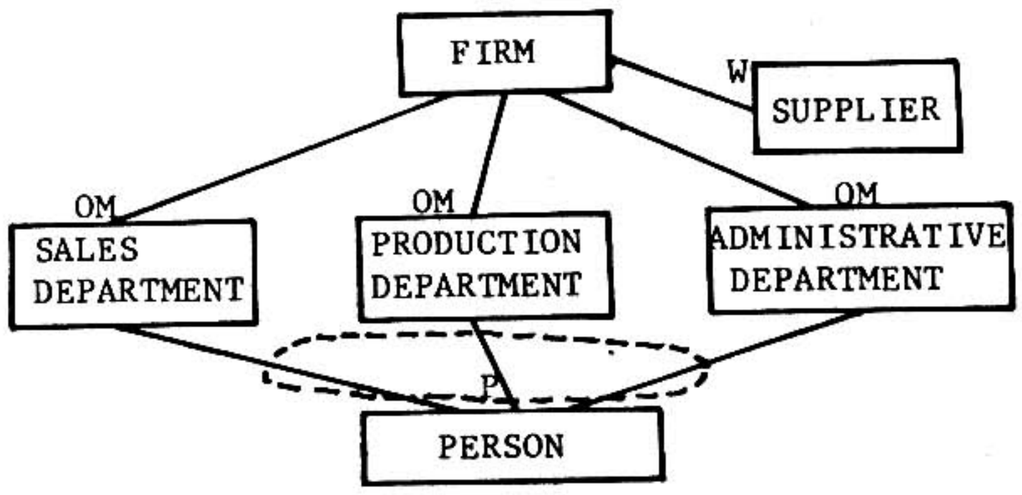
If A and B are two individuals connected by a relation R (A, B) we will say that B is an obligatory member of A with respect to R if
$b \in B \Rightarrow \exists\, a \in A : \{a, b\} \in R$.
If PERSØN (SURNAME, CHRISTIAN NAME, SEX) and ADDRESS (TOWN, NUMBER, STREET) are two individuals, the relation LOCALISATION (PERSØN, ADDRESS) represents the semantic relation of a person to his address ; we can say that ADDRESS is an obligatory member of LØCALISATIØN to represent the fact that a realisation of ADDRESS is necessarily the address of a person.

(ii) Member of a pool of individuals with respect to a pool of relations

An individual may "depend" on several relations and the preceding concept can be generalized in the following way :
If $A_1, A_2 \ldots A_n$ are individuals, B is a member of the pool of the individuals $A_1, A_2, \ldots A_n$ with respect to the relations $R_1 (A_1, B), \ldots R_n(A_n,B)$ if
$b \in B \Rightarrow \exists a_1 \in A, \{a_1, b\} \in R, \vee \ldots \vee \exists a_n \in A_n \{a_n, b\} \in R_n$

(iii) Weak member of an individual with respect to a relation

If A and B are two individuals connected by a relation R(A, B) we will say that B is a weak member of B with respect to R if B is not an obligatory member of R.



OM : obligatory member
W : weak member
P : pool

Fig. 2. an example of a data base

If PERSON and FIRM are individuals, PERSON is a weak member of the individual FIRM with respect to the relation EMPLOYEE (FIRM, PERSON) if one wants to show that the existence of a realization of FIRM is independent of the realizations of PERSON. A firm is, in effect, independent of the persons it employs.

2.4 Consequences of this typology

The properties of existence of individuals have important consequences for the operations of addition and suppression.

(i) Obligatory member

If A and B are individuals connected by the relation R (A, B) where B is an obligatory member of A with respect to R, we have the following properties:

- the set R_A is the domain of the individual B at any instant.
- If we add a realization to the individual B, it must be connected to at least one element of A.
- If we suppress an element of A, the realizations of B which are connected to it, must also be suppressed providing they are not connected to any other element of A.
- If we suppress a realization {a, b} of the relation R (A, B), the element $b \in B$ must be suppressed providing it is not connected to another element of $a \in A$.

We see then that the typology allows either :

- semantic controls, or
- automatic operations of addition or suppression.

(ii) Member of a pool

If B is a member of the pool of the individuals $A_1 \ldots A_n$ with respect to the relation $R_1(A_1,B)$, ..., $R_n(A_n B)$, then :

- The union of the sets $R_{i_{A_i}}$ i=1, ..., n, is the domain of B at any instant.
- If we add an element to the individual B it must be connected to atleast one element of atleast one of the individuals $A_1, \ldots A_n$.
- If we suppress a realization of one of the A_i $i \in [1,n]$, the realizations of B which are connected to it must also be suppressed, providing they are not connected to other elements of the A_i $i \in [1,n]$.

3. DATA BANK CONSULTATION LANGUAGE

The classic functions of a language for data banks are well-known : they are the consultation, updating (modification, addition and suppression) and definition of information. However, if we consider the case of a manager of a firm, for example, he asks himself certain questions of variable complexity, amongst which we can isolate certain semantic primitives which could form his personal language in fact the elementary functions of consultation and updating do not interest him in themselves as they are too restrictive for the problems he has to consider. Also, the personnel director, the manager of the stocks or the production manager will consult the firm's data bank in different ways, which means that each must have a language specific to the problems he has to treat, formed of special primitives which will be expressed in terms of elementary functions of the kernel language. This kernel language is based on the relational model, i.e. it contains commands which manipulate the entities considered in the latter. The first objective is simplicity ; we think that a conversational mode will allow us to reach this objective since it permits the system to intervene if the user has difficulty defining his request and it permits step-by-step working where the user does not ask his question at one go.

The kernel language in question comprises a specification language whose role is to designate one or more realizations of an individual, and a command language which allows one to specify the actions one wants to act on the designated individual.

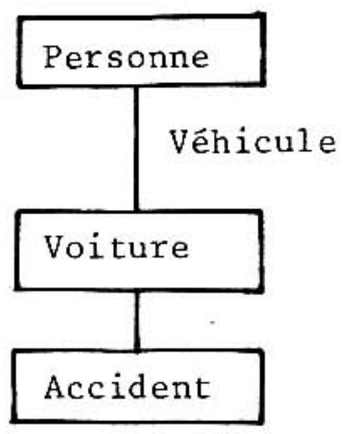


Fig. 3. Example of a data base

If we consider the base described in fig. 3 we can write the following request :
"Rechercher une personne ayant (nom Dupont et existe une résidence secondaire et un véhicule dont (marque Simca et n° 12B47 et un accident dont (date = 12/3/72)))".
This expression of a language consists of a key word "Rechercher" which indicates a command and a designation of the individual Person through the intermediary of a filter which gives a series of conditions on the individual. This filter is introduced by a key word, one of "ayant", "tel que", "dont", "qui a" and is a boolean expression of criteria of value, existence and relationship.

A criterion of value expresses a condition on the values of a characteristic of an individual and has the syntactic form
<characteristic> <operator> <value>
where the operator can be omitted if it is equality. "Nom Dupont" is a criterion of value, "existe une résidence secondaire" is a criterion of existence which tests that the characteristic "résidence-secondaire" has a value.

A criterion of relationship expresses the existence of an individual, possibly filtered, connected with the designated individual, for example "un véhicule dont (marque = Simca et ... date = 12/3/72).

<filtre> ::= <mot clé> (<condition>) ;
<mot clé> ::= tel que / ayant / qui a / ayant pour / a / dont ;
<condition> ::= <condition> [et <condition3>]$_0^\infty$;
<condition3> ::= <condit> [ou <condit>]$_0^\infty$;
<condit> ::= <condition1> / <condition2> ;
<condition1> ::= <caractéristique> [op]$_0^1$ <valeur> ;
<condition2> ::= <quantificateur> <relation> <filtre> / existe <caractéristique> / existe aucun <caractéristique> ;
<op> ::= = / ≠ / > / ≥ / < / ≤ ;
<quantificateur> ::= un / tous / N / premier / dernier / NAME ;

Syntax of a filter

Since an individual possesses a variable number of realizations, we use a quantifier to specify the number of realizations that interest us.

The command language contains essentially the following actions :

- access to one or more realizations of an individual.
- modification of values of one or more characteristics of an individual.
- addition of a realization of an individual.

- suppression of one or more realizations of an individual.
- conditional statements.
- grouping of actions.
- addition of a realization of a relation.
- suppression of a realization of a relation.
- creation of an individual.
- creation of a relation between two individuals.

4. CONCLUSION

At the present time, we are at the implementation stage and it is, therefore, out of the question to come to any final conclusions. The main problem in data structures is to represent in a more or less natural way or as faithfully as possible the information which the user wishes to store and manipulate in the data base. In this paper an attemp has been made to resolve this problem by introducing the concepts of individual and relation and by analyzing the properties of these concepts which could be interesting for the representation according to the criteria which we have defined. We have also presented our ideas for a consultation language for data banks. In addition to the classic functions which we consider as elementary, we have to provide the users with a language of semantic primitives. This is the aim of our current work.

APPENDIX

Let A and B two sets.
We define a relationship as a subset of the set of all the pairs we can form by taking an element of A and an element of B.

We define de following sets :

$a \in A \quad R_A[a] = \{b \ / \ b \in B \wedge \{a, b\} \in R\}$

$b \in B \quad R_B[b] = \{a \ / \ a \in A \wedge \{a, b\} \in R\}$

$$R_A = \bigcup_{a \in A} R_A[a]$$

$$R_B = \bigcup_{b \in B} R_B[b]$$

REFERENCES

[1] E.F. CODD, A Relational Model of Data for Large Shared Data Banks, Comm. ACM 13, 6, June 1970, 377-387.

[2] E.F. CODD, A Data Base Sublanguage founded on the Relational Calculus, IBM Research Report RJ 893, San Jose, California, July 26, 1971.