RELATIONAL ALGEBRA REVISITED:
TOWARD AN OBJECT-PREDICATE ORIENTED
RELATIONAL MODEL.

by

Victor Markowitz

Technical Report #319
June 1984
RELATIONAL ALGEBRA REVISITED: TOWARD AN OBJECT-PREDICATE ORIENTED RELATIONAL MODEL

Victor M. Markowitz, Technion, Israel

ABSTRACT

The domain-oriented approach to the Relational Model brought its data description side closer to the way people perceive information, that is, following an object-predicate pattern. Accordingly, one would expect that the model’s manipulative part would reflect the way people communicate, namely by natural language.

To this end we propose to modify the Relational Algebra in order to make each of its operations to have a linguistic analog. Moreover, the construction of complex algebraic expressions will follow patterns of natural language sentence combination. The change preserves the expressive power of the original Relational Algebra and makes the relational manipulation easier due to its new, natural language, orientation.

Categories and Subject Descriptors: H.2.3[Database Management]: DML; query languages.

General Terms: Human Factors, Languages

Additional Key Words and Phrases: Relational Model, Relational Algebra.
1. INTRODUCTION

When the Relational Model was proposed by Codd, one of its main objectives was communicability, namely to keep the model simple enough so that the users could easily understand, use and communicate with one another about the data [5]. This objective was only partially fulfilled by the model. Relational query languages, although higher level and easier to use than other query languages, have been found hard to manage by casual users; thus, for instance, comprehension difficulties, for some of them, have been pointed out by human factors studies such as [10] and [12]. The structural flatness of the initial form of the model was at fault generally. It favored semantic misunderstandings and the expression of syntactically correct queries that were not reflecting the user's intended meaning. Moreover, the user was compelled to adapt himself to a simple, but not necessarily natural, view of the real world.

In Section 2 we shall review the domain oriented version of the Relational Model and its linguistic interpretation. The domain orientation given to the model (e.g. [9]) has been successful in its attempt to bring the data description closer to the natural perception of information. Accordingly, one would expect that the model's manipulative part would draw closer to the natural way of communication, the natural language. To this effect a reshaping of the Relational Algebra has been outlined in [8]; it has been a first attempt to change the Relational Algebra in the
sense of the modification proposed in Section 3. The linguistic analogies of this algebra and how this algebra relates to the original Relational Algebra, are also discussed in Section 3.
2. THE RELATIONAL MODEL

The following review of the basic relational structural concepts and of the relational algebra is based mainly on [9].

2.1 Relational Model: The Structural Part

A relational database consists of a set of relations. Its structural definition, called schema, includes the specification of a set of domains and a collection of relation schemes. A collection of consistency constraints is also part of the schema, but will be omitted as being irrelevant in the context of this paper.

A domain is a finite set of atomic elements (values). It has:

- a name which identifies it uniquely within the database;
- a collection of binary comparison operators including, at least, an equality test operator; and
- a, possibly empty, collection of operations for the elements of the domain.

The function (role) fulfilled by a domain in a relation is identified by a role.

A relation has a name, a structural description called relation-scheme, and a value. The relation-scheme (r-scheme) is
specified by a set of domain-role pairs. The domain-role specification is called attribute. The set of attributes corresponding to a relation, may be viewed as a set of distinct indexes, with every index, a, associated with a domain, Da.

The relation-value (r-value) of a relation is a subset of the indexed Cartesian product [9]:

\[
\{ t: A \rightarrow \cup Da \mid \forall a \in A: t(a) \in Da \}
\]

where the tuple \( t = < a_1: d_1, \ldots, a_k: d_k > \) is described as a total function on the set of attributes A, having values in the associated domains. Note that these labeled n-tuples are unordered sets of attribute-value pairs.

Every domain has associated a unary relation called an e-relation [4], whose purpose is to assert the existence of the objects belonging to that domain; it lists all the identifiers of the objects belonging to that domain, currently recorded in the database. Both the e-relation and the attribute of its r-scheme, are named after the respective domain; the corresponding role is null.

2.2 Relational Model: The Relational Algebra

Relations are manipulated by Relational Algebra (RA) operators. We shall use the following notations:
r, s, ... denote relations;
a, b, ... denote single attributes;
A, B, ... denote sets of attributes;
t(a) denotes the value corresponding to the attribute a, in the tuple t;

uv denotes a tuple obtained by the concatenation, meaning in the present context the union, of the tuples u and v; and

t[A] denotes a tuple containing components of t corresponding to the attribute set A.

An algebraic operation has one or two operands, which are relations, and its result is another relation, with its own r-scheme and r-value.

(1) **Attribute renaming.** The application of any algebraic operator may be preceded by the specification of an attribute correspondence, accomplished through the renaming of the attributes of the operators. The rename operation has the following form:

rename (r, f), where r denotes a relation whose attribute set is A and f is a one-to-one, not necessarily total, functional mapping \{(a->c)\}, with aEA.

As a result, r will have renamed attributes, such that every image of a under f, c, will be associated with the domain of a:
in the tuples of \( r \) every attribute from the domain of \( f(\cdot) \) is replaced by \( f(a) \). The result of algebraic operations will directly inherit the attributes from, possibly renamed, operands, and, on their turn, may be subjected to renaming and appear as operands in other operations.

(2) **Projection.** Given a relation \( r \) whose attribute-set is \( A \), the projection of \( r \) on a subset of \( A, A' \), is defined as:

\[ [A']r = \{ t[A'] | t \in r \}. \]

(3) **Cartesian product.** Given two relations, \( r \) and \( s \), whose respective attribute-sets, \( A \) and \( B \), are disjoint, their cartesian product is defined as:

\[ r \times s = \{ t \mid t = uv, u \in r \text{ and } v \in s \}. \]

(4) **Union, Difference, Intersection.** Let \( A \) and \( B \) be two sets of attributes corresponding to the relations \( r \) and \( s \) respectively; \( r \) and \( s \) are said to be union-compatible if there is a total one-to-one correspondence between \( A \) and \( B \), such that corresponding attributes in \( A \) and \( B \) are associated with a same domain. Given two union-compatible relations, \( r \) and \( s \), with the attributes of \( s \) renamed as their correspondents in \( r \), the union/difference or intersection of \( r \) and \( s \) are the known set operations taking \( r \) and \( s \) as operands.

Interesting and useful generalizations of the above operations are their bordered counterparts [9], which accept not
necessarily union-compatible operands. Let \( r \) and \( s \) be two relations whose attribute-sets are \( A \) and \( B \) respectively, and \( C = A \cap B; \ A' = A - C; \ B' = B - C. \) The bordered union (intersection) involving \( r \) and \( s \) is the ordinary union (intersection) involving two union-compatible relations: one is obtained by bordering each tuple of \( r \) by all the possible values for the attributes of \( s \) that are not of \( r, B' \); and the other is obtained by bordering each tuple of \( s \) by all the possible values for the attributes of \( r \) that are not of \( s, A' \). The borderings are accomplished through the cartesian product of \( r, \) respectively \( s, \) with the \( e \)-relations, renamed if necessary, corresponding to every attribute of \( B', \) respectively \( A'. \)

(5) **Restriction.** Given a relation \( r \) and an atomic comparison of the form \((a \approx K)\) or \((a \not< b)\), where \( a \) and \( b \) both belong to the attribute-set of \( r, B \) is in the collection of comparison operators associated with the domain of both \( a \) and \( b, \) and \( K \) is a constant, a restriction applied on \( r \) has the following forms:

\[
\begin{align*}
 r[a \approx K] &= \{ t | t \in r \text{ and } t(a) = K \}, \text{ or } \\
 r[a \not< b] &= \{ t | t \in r \text{ and } t(a) < t(b) \}.
\end{align*}
\]

(6) **Natural Join.** Given two relations, \( r \) and \( s, \) \( A \) and \( B \) their respective attribute-sets, and \( C = A \cap B, \) the natural join of \( r \) and \( s \) is defined as:

\[
 r \bowtie s = \{ t | t = u[y[B-C]], u \in r, v \in s \text{ and } u[C] = v[C] \}.
\]
Notice that the natural join is equivalent to the bordered intersection.
(7) **G-Join.** Given two relations, \( r \) and \( s \), such that \( A \) and \( B \), their respective attribute-sets, are disjoint, and \((a \& b)\) a comparison as described above (5), the G-Join of \( r \) and \( s \) on \((a \& b)\) is defined as:

\[
r[a \& b]s = \{ t \mid t = uv, \ u \in r, \ v \in s \text{ and } u(a) \& v(b) \}.
\]

(8) **Generalized division.** Given two relations, \( r \) and \( s \), \( A \) and \( B \) their respective attribute-sets, and \( C = A \cap B \), the division of \( r \) by \( s \) is defined as:

\[
r/s = \{ t \mid t = uv, \ u \in r[A-C], \ v \in s[B-C] \text{ and }\langle w[C] \mid w \in r, \ w[A]=u[A] \text{ contains } \langle w[C] \mid w \in s, \ w[B]=v[B] \rangle \}.
\]

This is a generalization of the division defined in [3].

Compared to Codd's original definition of the Relational Algebra [3], the algebra presented above has the same selective power, that is, it is relationally complete (see [9] for details); moreover, its additional, more complex, operators make it easier to use.

2.3 The Linguistic Interpretation of the Relational Model

The object-predicate perception of information is supported by choosing user oriented domains; where every domain groups objects of a same type; the domains are correlated in a relation only if the relation models a meaningful association in the framework of the enterprise. The relations will represent relationships.
stating the fact that objects interact, or properties, that are
describing objects. All the objects of a same type, that is
belonging to a same domain, share the same properties, and we
assume that the existence of an object implies the existence of
all the properties associated with the domain it belongs to.

Within the database the domain elements need unique and permanent
identifiers. Objects that are aggregates of other objects are
represented by system assigned surrogates (see [4] for a detailed
discussion on this topic). A surrogate is information free, and
protected from the user who may do no more than cause the system
to delete or generate a surrogate. A domain of surrogates has
associated only an equality test operator: two surrogates are
equal if, and only if, they denote the same entity in the
perceived world of entities. Notice that the user declares
whether two entities are distinct.

The domains may be denoted by 'real-world' noun names. Likewise,
since r-value tuples model facts expressed by natural language
sentences, every relation may be associated with the sentence
type of these sentences [1]. We assume that the sentences are
elementary, that is, consist of a single verb and one or several
noun phrases, where every noun denotes a domain. We shall use
these sentence types as relation names; since every sentence
type has different paraphrases, the corresponding relation will
have several denotations. A SUPPLY relation, for instance, could
be denoted by:
ITEM is SUPPLIED to DEPARTMENT by SUPPLIER;
DEPARTMENT is SUPPLIED with ITEM by SUPPLIER; and
SUPPLIER SUPPLIES ITEM to DEPARTMENT.

What differentiates the paraphrases is which of the terms is in
the subject position, while the ordering of the terms in the
object position is irrelevant. The predicate of the paraphrase
in which a specific domain is in the subject position,
characterises the function of that domain in the relation;
therefore we choose it to be the role of the domain in the
corresponding r-scheme. The r-scheme of the SUPPLY relation
above, for instance, is:

< DEPARTMENT: SUPPLIED; ITEM: SUPPLIED; SUPPLIER: SUPPLIES >.

E-relations are described by sentence types of the form "be
domain name", stating the existence of sets of objects, the
domains, as opposed to the existence of associations.

Following the Closed World Assumption of [11], the sentences
represent only positive facts, while negative facts are
considered as being implied by the lack of any positive
counterpart within the database.

Had we been using a graphical technique to represent the schema,
like the Chen's Entity-Relationship Diagram [2], we would obtain
the surface structure of the sentence types associated with the
relations of the database.

The domain oriented definition of the RM structural part does not
emphasize the table view of relations and offers a more intuitive way of perceiving the relations, as modelling an object-predicate semantics. This view, closer to the way people perceive information, both favors and suggests the definition of a manipulative part drawing on analogies with the way people communicate, the natural language.

Another step toward an object-predicate oriented Relational Model would be to base the design process on the natural language sentences describing the modelled enterprise. To this end Biller has proposed semantic-irreducibility [1] as a goal for relational normalization: a relation is said to be semantic-irreducible if it models an elementary sentence type, that cannot be split into conjunctions of other sentence types. Similar trends are also followed by recent works in the field of relational database design (e.g. [6]).
3. AN OBJECT-PREDICATE ORIENTED RELATIONAL ALGEBRA

In this section we propose a modification of the Relational Algebra. The object-predicate oriented algebra, denoted OP/RA, takes advantage of the possibilities posed by the linguistic analogies of the RM structural part, namely the relation denotations based on natural language sentence types, by using constructs similar to the natural language sentence combination. In the above presentation of the Relational Algebra we have paid attention only to its elementary operations. It is known that RA is closed under composition, meaning that an algebraic expression can be used as operand in any other algebraic operation. When embarking on the redefinition of the RA, we had in mind the construction of such complex expressions which we intend to make follow patterns of natural language sentence composition.

We call an algebraic expression a query. A query denotes a derived relation with neither a structure declared in the schema, nor a simple name, and with an r-value produced by an evaluation process based on the semantics of the algebraic operations; we call such relations query-relations (q-relations). A q-relation is described by a complex sentence type involving the elementary sentence types corresponding to the relations referenced within the query. For every OP/RA operation we shall establish its linguistic analog, that is we shall investigate its relationship to the rules governing the combination of these elementary sentence types. We shall refer to the following combination
patterns (a detailed presentation may be found in [13]):

- **relativization**, meaning the connection of a sequence of sentences such that any two neighbouring sentences are chained by a relativizer, such as that or which, on an object term. There are two main forms:

  - **restrictive relativization**, where the connecting object term is raised from an object position in the first sentence, to the subject position in the second sentence; the relativizer need not be present. For instance:
    
    "department requesting item supplied by supplier";

  - **non restrictive relativization**, where the connecting relativizer is not standing for the subject of the second sentence in a chair pair. For instance, the above example may be reformulated as:
    
    "item requested by department to which some supplier supplies this item";

- **coordination**, meaning the connection of several sentences with the help of the logical connectives and and or. For instance:

  "item requested by department and supplied by some supplier";

We shall call the relations that are declared in the schema and have a permanently maintained r-value regular-relations.
(r-relations). The schema comprises the specification of the roles of the domains in the various relations they are involved in. Within a query one’s concern shifts toward expressing the correlations of the various appearances of the domains in that query. This change is reflected by the structure of the operand-relations that will function as the operands of algebraic expressions; the operand-relation (o-relation) is a temporary relation, derived from an r-relation, meaningful only within a particular query. The correlations are assured by identifiers, called correlators. A domain is associated within a given query q with a set of correlators, such that a correlator correlates several occurrences of that domain in different o-relations of q.

The structure of an o-relation is described by an operational-scheme; the operational-scheme (o-scheme) of an o-relation consists of a set of pairs domain-correlator. A domain-correlator specification is called operational-attribute (o-attribute). A subset of the set of all the o-attributes referenced in a query specifies the target of the query, that is the o-scheme of the corresponding q-relation.

We are using the following notations:

- \( rR \) denotes an r-relation;
- \( eR \) denotes an e-relation;
. $sR$ denotes an $o$-relation;

. $a, b, \ldots$ denote single attributes of $r$-relations;

. $t(a)$ denotes the value corresponding to the attribute $a$, in the tuple $t$;

. $uv$ denotes a tuple obtained by the concatenation, meaning in the present context the union, of the tuples $u$ and $v$;

. $A, B, \ldots$ denote sets of attributes of $r$-relations;

. $t[A]$ denotes a tuple containing components of $t$ corresponding to the attribute set $A$;

. if $rR$ is a relation and $a$ is an attribute belonging to the attribute set of $rR$, then $\sum(a;rR) = a: \sum_{t\epsilon rR} t(a)$;

. $x, y, \ldots$ denote single $o$-attributes;

. $\{x\}$ denotes a set of values labeled by $x$;

. $\text{card}(x)$ denotes the cardinality of the set $\{x\}$;

. $X, Y, \ldots$ denote sets of $o$-attributes;

. $\text{CARD}$ denotes a domain containing positive integer numbers; and
S(R) denotes the o-scheme of sR.

The query examples in this section are expressed in a slightly modified version of ERROL (Entity Relationship Role Oriented Query Language)[7]. Roughly, ERROL is derived from a natural language subset, by enforcing the observation of some restrictions. Domain names, roles and correlators are the ERROL identifiers; they appear written with upper case letters. Comments may be placed anywhere within a query and appear written with lower case letters. Whenever there is a need for a role that is not declared in the schema, a pseudo-role, HAVE, may be used; HAVE asserts the existence of a relationship without expressing what that relationship is. The target of an ERROL query is stated in a GET-CLAUSE, headed by the key-word get and consisting of a list of one or more attributes separated by colons, for instance:

```
get SUPPLIER, ITEM:1 ...
```

An ERROL query defines a q-relation, and the GET-CLAUSE asserts its o-scheme; whenever the correlators are not explicitly stated, there are provided by default. Thus, the o-scheme corresponding to the above GET-CLAUSE is: {SUPPLIER:1; ITEM:1}. An ERROL query has, beside a GET-CLAUSE, one or several TIS-CLAUSEs asserting the associations connecting the target domains. A TIS-CLAUSE is delimited by one of the key-words that, which or at (such that) and a semicolon, and involves relation denotations and atomic comparisons, combined by relativization or coordination, in a complex qualification phrase. The following
implicit referencing is used, whenever possible, in order to keep the referencing limited:

- a special correlator, "*", associated with a domain within the TIS-CLAUSE means a reference to the corresponding, uncorrelated, domain of the GET-CLAUSE; and

- two o-attributes within the TIS-CLAUSE, associated with a same uncorrelated domain, are implicitly different.

In natural language complex sentences, the various appearances of a same noun in different component elementary sentences are correlated with the help of determiners such as this, that, etc. An implicit correlation is provided by textual contiguity. The correlators fill, both in OP/RA and EROL, the referencing task of these determiners and are easily deduced from the natural language sentence referencing. For instance

"department requesting item supplied to this department", is interpreted as

DEPARTMENT:D REQUESTING ITEM: I SUPPLIED to DEPARTMENT:D.

Let \( sR' \) and \( sR'' \) be two o-relations referenced within a query; the common part of their o-schemes is their set of mutual references:

\[
M(sR', sR'') = S(sR') \cap S(sR'').
\]
(1) **Renaming.**

The o-relations are derived from r-relations by a set of renamings. Let \( R \) be an r-relation. \( A \) its set of attributes, and \( x \) a set of o-attributes; \( f \) is a one-to-one, not necessarily total, mapping associated with \( R \), \( \{(a \rightarrow x)\} \), where \( o \) is either \( \text{sum} \), \( \text{card} \), or the empty string, and \( a \in A \), \( x \in X \), for \( o \) being \( \text{sum} \) or the empty string, both \( a \) and \( x \) are associated with a same domain, while for \( o \) being \( \text{card} \), \( x \) is associated with \( \text{CARD} \). Whenever \( o \) is \( \text{sum} \) or \( \text{card} \), the corresponding o-attribute is called computed o-attribute. The rename operation is defined as follows:

\[
s_R = \text{rename}(R, f) = \{ t : X \rightarrow U \mid \exists t' \in R \text{ such that:} \\
\forall (a, x) \in f, \quad t(x) = t'(a), \\
\forall (\text{sum}, x) \in f, \quad t(x) = \text{sum}(a; \{ w[A-A'] \mid w \in R, \ w[A'] = t'(A') \}), \quad \text{and} \\
\forall (\text{card}, x) \in f, \quad t(x) = \text{card}(w[a] \mid w \in R, \ w[A'] = t'(A')) \},
\]

where \( A' = \{ a \mid (a, x) \in f \} \).

The domain of \( f \) is a subset of \( A \), while the range of \( f \) is the o-scheme of \( s_R \), \( S(s_R) \); the renaming operation derives the r-value of \( s_R \) by performing an index replacement in \( rR[A'] \), together with value computations for the computed o-attributes.

The OP/RA renaming is applied only on r-relations, and the renamings associated with a query provide all the o-relations referenced in that query.

Every r-relation is described by, and associated with, a
elementary predicative sentence type whose object terms denote the domains involved in the relation. When one or several of the object terms are omitted we are dealing, actually, with an implied, new sentence type describing a new association represented by the o-relation resulting from the renaming applied on that r-relation. Hence the o-relations produced by renamings have simple names derived from the names of the r-relations. The rename mapping associated with the r-relation and the resulting o-relation are straightforwardly deduced from the ERROL expressions. For example,

DEPARTMENT SUPPLIED with ITEM,

denotes an o-relation sR, whose o-scheme is

\[ S(sR) = \{ \text{DEPARTMENT:D}, \text{ITEM:2} \} \]

and

DEPARTMENT:D SUPPLIED with \text{card} ITEM,

denotes an o-relation sR, whose o-scheme is

\[ S(sR) = \{ \text{DEPARTMENT:D}, \text{CARD:2} \} \]

Unlike ordinary o-attributes, computed o-attributes have only a local significance, therefore they cannot be explicitly correlated.

A query is evaluated in stages, where every stage covers one of the OP/RA operations defined below.

(2) Projection.

Let sR' be an o-relation, \( X \) a subset of \( S(sR') \), \( Y \) a set of
o-attributes and \( g \) a one-to-one, not necessarily total, mapping associated with \( sR' \), \( \{(x \rightarrow y)\} \), where \( g \) is either \text{sum} \text{, card} \text{, or the empty string} \text{, and } x \in X \text{, } y \in Y \text{, for } g \text{ being sum or the empty string} \text{, } y \text{ is associated with the same domain as } x \text{, while for } g \text{ being card} \text{, } y \text{ is associated with CARD} \text{. Moreover, for every } (x \rightarrow y) \in \text{Eg} \text{, } y=x; \text{ the common part of } X \text{ and } Y \text{ is } X'=X \cap Y=\{x1 \in \text{Eg} \} \text{.}

The projection of \( sR' \) is defined as follows:

\[
sR=\{ [g]sR'=\{ t : \forall y \rightarrow \text{UDy} \exists t \in sR' \text{ such that: } y \in Y \}
\]

\[
\forall (x,y) \in \text{Eg}, \ t(y)=t'(x),
\]

\[
\forall (\text{sum},y) \in \text{Eg}, \ t(y)=\text{sum}(x,\{w[S(sR')-X']\} \text{wEsR'}, w[X']=t'[X'])
\]

and \( \forall (\text{card},y) \in \text{Eg}, \ t(y)=\text{card}(w[x] \text{wEsR'}, w[X']=t'[X']) \) \text{.}

Notice that when \( Y=X \text{, the above definition is equivalent to that of the RA projection} \text{.}

The OP/RA projection is mainly meant to provide a way of explicitly stating the structure of an evaluated o-relation that is not the result of an OP/RA renaming. In ERROL a reference to such an o-relation is marked by the pseudo-role and the utilization of one of the determiner key-words that, which or st.

For example,

\text{SUPPLIER:S HAVING ITEM that is,}

\text{REQUESTED by a DEPARTMENT SUPPLIED by SUPPLIER:S;}

explicitly refers to an o-relation whose o-scheme is \{SUPPLIER:S; ITEM:\}. Likewise, the declaration of the o-scheme of the q-relation in the GET-CLAUSE, states an OP/RA projection.
Thus,

\texttt{get DEPARTMENT, ITEM at \ldots}

implies a projection on \{DEPARTMENT: 1, ITEM: 2\}.

The OP/RA projection provides the means of explicitly referring derived o-relations not only of relationship type, but also of property type. Let, for example, \(sR\) and \(sR''\) be denoted by \texttt{SUPPLIER:S SUPPLYING DEPARTMENT and DEPARTMENT REQUESTING ITEM in GTY, respectively}, then \texttt{SUPPLIER:S HAVING \texttt{sumGTY} that is REQUESTED of some ITEM by DEPARTMENT SUPPLIED by SUPPLIER:S; denotes \{SUPPLIER:S \texttt{sumGTY} (sR' \bigcap sR'')\}.}

(3) Cartesian product.

Let \(sR\) and \(sR''\) be two o-relations. Their cartesian product is defined as follows:

\[ sR = sR' \times sR'' = \{(t, t) \mid u \in sR' \text{ and } v \in sR''\}, \quad \text{and} \]
\[ S(sR) = S(sR') \cup S(sR''). \]

The cartesian product is not explicitly used in queries, it is only embedded in some of the other OP/RA operations, as shown below.

(4) Intersection, Union, Complement.

Let \(sR\) and \(sR''\) be two o-relations whose o-schemes do not involve computed o-attributes. The OP/RA intersection and union
are similar to their RA bordered counterparts. Both are preceded by the following bordering sequence:

- \( sR' = sR' \) and \( sR'' = sR'' \);

- for every \( o \)-attribute, \( x \), in \( S(sR') - M(sR', sR'') \), corresponding to a domain associated with an \( e \)-relation \( eP \) whose single attribute is \( d \), perform:
  
  \[
  sP = \text{rename}(eP, (d, x)) \quad \text{and} \quad \text{\( sR' \star \text{rename}(eP, (d, x)) \)}
  \]
  
  \[
  sR'' = sR'' * sP, \quad S(sR'') = S(sR') US(sP) \]

- for every \( o \)-attribute, \( x \), in \( S(sR') - M(sR', sR'') \), corresponding to a domain associated with an \( e \)-relation \( eP \) whose single attribute is \( d \), perform:
  
  \[
  sP = \text{rename}(eP, (d, x)) \quad \text{and} \quad \text{\( sR'' \star \text{rename}(eP, (d, x)) \)}
  \]
  
  \[
  sR' = sR' * sP, \quad S(sR') = S(sR'') US(sP) \]

Consequently, the \( DP/RA \) intersection and union of \( sR' \) and \( sR'' \) are defined as follows:

- \( sR = sR' \cup sR'' = \{ t \mid t \in sR' \text{ or } t \in sR'' \} \), and
  
  \[
  sR = sR' \cap sR'' = \{ t \mid t \in sR' \text{ and } t \in sR'' \} \]

- where \( S(sR) = S(sR') US(sR'') \).

Given an \( o \)-relation \( sR' \), the complement of \( sR' \) is defined as follows:
Let $sR'$ and $sR''$ be two o-relations denoted by two sentence types; their combination by relativization denotes the $OP/RA$ intersection of $sR'$ and $sR''$. Remark that the textual contiguity characteristic to restrictive relativization, means a non empty $M(sR', sR'')$. For example,

ITEM: I REQUESTED by DEPARTMENT SUPPLIED with ITEM: I, is represented by

$$sR' \bigcap sR'', M(sR', sR'') = \{DEPARTMENT: 1, ITEM: I\},$$

where $sR'$ and $sR''$ are denoted by

ITEM: I REQUESTED by DEPARTMENT with

$S(sR')=\{ITEM: 1, DEPARTMENT: 1\}$, and

ITEM: I SUPPLIED to DEPARTMENT with

$S(sR'')=\{ITEM: 1, DEPARTMENT: 1\}$, respectively.

And coordination also corresponds to $OP/RA$ intersection. Notice that when the connected sentences have a common subject, the and
coordination is equivalent to restrictive relativization. The above OP/RA intersection, for example, may also be denoted by DEPARTMENT REQUESTING ITEM: I and SUPPLIED with ITEM: I.

The OP/RA complement corresponds to the Closed World Assumption [11] negation, that is, expressing facts whose positive counterparts are not explicitly present in the database. Thus, for example, if $sR'$ is denoted by DEPARTMENT REQUESTING ITEM, then

DEPARTMENT not REQUESTING ITEM, denotes

not $sR'$.

The complement is derived from, and replaces, the RA bordered difference which can be now expressed as: $sR' \n (not \ sR'')$; the natural language correspondent of difference is the and not coordination.

Finally, or coordination corresponds to OP/RA union. Thus, for instance,

DEPARTMENT REQUESTING some ITEM or SUPPLIED with some ITEM is represented by:

$sR' \cup sR''$, where $M(sR', sR'') = \{DEPARTMENT:1\}$.

(5) Restriction.

Given an o-relation $sR$ and an atomic comparison of the form $(x \theta K)$ or $(x \theta y)$, where $x$ and $y$ are both o-attributes of $sR$, $\theta$ is
in the collection of comparison operators associated with the
domain of both $x$ and $y$, and $K$ is a constant, an OP/RA restriction
applied on $sR'$ has the following forms:

$$sR = sR'[x \leq K] = \{ t \mid t \in sR', \text{and } t(x) \leq K \}, \text{ or}$$

$$sR = sR'[x \geq y] = \{ t \mid t \in sR', \text{and } t(x) \geq t(y) \} \text{, where}$$

$$s(sR) = s(sR').$$

The OP/RA restriction corresponds to the chaining by
relativization of a sentence type with a comparison statement.

Let, for instance, the $o$-relation $sR'$ be denoted by

DEPARTMENT REQUESTING ITEM in QTY, with

$$s(sR') = \{ \text{ITEM:1; DEPARTMENT:2; QTY:2} \}; \text{ then}$$

DEPARTMENT REQUESTING ITEM in QTY > 10, denotes

$$sR'[\text{QTY:2>10}].$$

When the comparison involves a computed $o$-attribute associated
with CARD, the OP/RA restriction corresponds to the use of
natural language quantifiers of the form 'at least $K$', 'at most
$K$', etc. For example, let $sR'$ be denoted by

DEPARTMENT REQUESTING ITEM, with

$$s(sR') = \{ \text{ITEM:1; DEPARTMENT:2} \}; \text{ then}$$

DEPARTMENT REQUESTING $\text{card ITEM} > K$; that is,

"department requesting at least $K$ items", denotes

$$sR'[\text{CARD:2} > K].$$

(6) $B$-Join.

Let $sR'$ and $sR''$ be two $o$-relations, and $(y_0z)$ a comparison,
where \( y \in \text{ES}(sR')\), \( z \in \text{ES}(sR'')\), both \( y \) and \( z \) are associated with a same domain, and neither \( y \) nor \( z \) belongs to \( M(sR', sR'')\). The OP/RA \( 0 \)-join of \( sR' \) and \( sR'' \) on \((y \& z)\) is defined as follows:

\[
sR = sR' \cdot (y \& z), sR'' = \{ t \mid t = uv[M(sR', sR'')], \]

\[
u \in sR', v \in sR'', u(y) = v(z), \text{ and } u[M(sR', sR'')] = v[M(sR', sR'')], \]

\[
B(sR) = B(sR') \cup U(sR''), \]

Two sentence types may be chained by modifiers such as ‘more than’, ‘same as’, etc., that is, on two object terms involved in a comparison; this kind of combination corresponds to the OP/RA \( 0 \)-join. Let for instance \( sR' \) and \( sR'' \) be denoted by

DEPARTMENT REQUESTING ITEM: \( I \) in QTY, with

\[
S(sR') = \{ \text{ITEM} : I; \text{DEPARTMENT} : 1; \text{QTY} : 2 \}, \] and

ITEM: \( I \) SUPPLIED by SUPPLIER in QTY, with

\[
S(sR'') = \{ \text{ITEM} : I; \text{SUPPLIER} : 2; \text{QTY} : 4 \}, \] respectively. Then,

DEPARTMENT REQUESTING ITEM: \( I \) in QTY >

QTY SUPPLIED of ITEM: \( I \) by some SUPPLIER, denotes

\[
sR' \cdot \text{QTY} : 2 > \text{QTY} : 4 sR'', \] where \( M(sR', sR'') = \{ \text{ITEM} : I \} \).

The above modifiers may also have a counting sense; then ‘more than’ means ‘\( \text{card-} > \text{card-} ' \), ‘same as’ means ‘\( \text{card-} = \text{card-} ' \), etc. For example, let \( sR' \) and \( sR'' \) be denoted by

DEPARTMENT REQUESTING ITEM, with

\[
S(sR') = \{ \text{ITEM} : I; \text{DEPARTMENT} : 2 \}, \] and

SUPPLIER SUPPLYING ITEM, with

\[
B(sR'') = \{ \text{ITEM} : 3; \text{SUPPLIER} : 4 \}, \] respectively. Then,
DEPARTMENT REQUESTING cardITEM >

(cardITEM SUPPLIED by some SUPPLIER; meaning

"department requesting more items than

supplied by supplier", denotes

sR'[CARD: 5 > CARD: 6]sR''.

(7) Set-join.

Let sR' and sR'' be two r-relations, and x and y two r-attributes belonging to S(sR') and S(sR'') respectively, such that both are associated with a same domain and neither x nor y belong to M(sR', sR''); \{(x)@\{y\}\} is a set comparison, where @ is a set comparator. The set-join of sR' and sR'' on \{(x)@\{y\}\} is defined as follows:

\[
sR = sR'[\{(x)@\{y\}\}sR'' = \{t \mid t = u[S(sR') - x]v[S(sR'') - y], u \in sR', v \in sR'', u[M(sR', sR'')] = v[M(sR', sR'')]\} and \\
\{w(x)\mid w \in sR' \text{ and } w[S(sR') - x] = u[S(sR') - x]\} \subseteq \\
\{w(y)\mid w \in sR'' \text{ and } w[S(sR'') - y] = v[S(sR'') - y]\} \\
S(sR) = S(sR') \cup S(sR'') - x - y.\]

Various universal quantifiers are used in natural language, for instance, 'all', 'at least', 'only', 'at most', etc. In the subsequent examples we shall replace them with set expressions which are more restricted in form, but also more accurate. Thus 'all' is expressed as 'set-contains-set', 'only' as 'set-equal-set', 'at most' as 'set-in-set' and 'at least' as 'set-contains-set'. We shall discuss only the 'set-contains-set'
form; as the other cases are similarly treated.

A set-comparison, just like a \( \Theta \)-comparison, can chain two sentences and such a chaining corresponds to the OP/RA set-join. For example, let \( sR' \) and \( sR'' \) be denoted by

\[
\text{DEPARTMENT: D REQUESTING ITEM, with } S(sR') = \{ \text{DEPARTMENT: D; ITEM: 1} \}, \text{ and }
\]

\[
\text{ITEM SUPPLIED by some SUPPLIER to DEPARTMENT: D with } S(sR'') = \{ \text{DEPARTMENT: D; SUPPLIER: 2; ITEM: 3} \}, \text{ respectively.}
\]

Then,

\[
\text{DEPARTMENT: D REQUESTING set ITEM contains}
\]

\[
\text{set ITEM SUPPLIED by SUPPLIER to DEPARTMENT: D}
\]

is represented by

\[
sR' \ast \{ \text{ITEM: 1} \} \ast \{ \text{ITEM: 3} \} \ast sR'', \text{ where}
\]

\[
M(sR', sR'') = \{ \text{DEPARTMENT: D} \}.
\]

For more complex quantifier expressions the nesting of such set-joins reflects the nested form of the natural language expression. For instance:

"get the suppliers supplying all the departments that are requesting some item, such that they supply every department with all the items the department requests";

may be split into

\[\text{(i) "get the pairs of suppliers and departments, such that a department is supplied by the respective supplier with all the items the department requests", corresponding to}\]

Technion - Computer Science Department - Technical Report CS0319 - 1984
get SUPPLIER, DEPARTMENT that is SUPPLIED by SUPPLIER:* with set ITEM contains set ITEM REQUESTED by DEPARTMENT:*

which denotes

\[ s_R = s_R\{\{\text{ITEM:1}\} \text{ contains } \{\text{ITEM:2}\}\} s_R', \text{ where} \]
\[ M(s_R, s_R') = \{\text{DEPARTMENT:3}\}; \text{ and} \]

(2) "get the suppliers associated in (1) with all the departments requesting something", corresponding to

get SUPPLIER that is HAVING

set DEPARTMENT that is related to supplier in (1);

contains set DEPARTMENT REQUESTING some ITEM;

which denotes

\[ s_R\{\{\text{DEPARTMENT:3}\} \text{ contains } \{\text{DEPARTMENT:4}\}\} s_R''', \text{ where} \]
\[ s_R''' \text{ is denoted by DEPARTMENT REQUESTING some ITEM and} \]
\[ M(s_R, s_R''') = \emptyset. \]

Putting together (1) and (2), the whole query is:

get SUPPLIER that is SUPPLYING set DEPARTMENT:D that is

SUPPLIED by SUPPLIER:* with set ITEM contains

set ITEM REQUESTED by DEPARTMENT:D; contains

set DEPARTMENT REQUESTING some ITEM.

Some of the OP/RA operations are derived from their RA homonyms, whose meaning they generally preserve. The OP/RA operators are distinguished from their RA counterparts by the special significance given to the o-schemes and the sets of mutual references. The OP/RA restriction, OP/RA union and intersection
are essentially the same operations as the RA restriction, bordered union and bordered intersection. The OP/RA complement is derived from, and replaces, the RA bordered difference. The OP/RA renaming combines the RA renaming and projection with a computed attribute derivation component. The OP/RA cartesian product is embedded in OP/RA union, intersection and complement, and has no utilization as an explicit operation. The OP/RA projection has been extended with a computed attribute derivation component. For the RA \( \cap \)-join the attribute-sets of the operands have to be disjoint. This condition is removed in OP/RA where the set of mutual references may be non empty. OP/RA \( \cap \)-join is a combination of RA \( \cap \)-join, natural join and renamings. The RA natural-join is equivalent to OP/RA intersection. Finally, the OP/RA set-join replaces the generalized division and it is a combination of generalized division on a single attribute, natural-join and renamings. We have defined the set-join such that it would reflect the form of natural language quantifier expressions. Finally, notice that the nesting in OP/RA reflects the natural language sentence apposition.
4. CONCLUSION

The domain oriented definition of the Relational Model's structural part has represented a first step toward an object-predicate oriented Relational Model. The modification of the Relational Algebra proposed in this paper is a further step taken in the same direction. The operations of the OP/RA reflect natural language sentence combination patterns, thus being closer to the natural way of communication. These operations require the special environment of the o-relations in which the stress is put on expressing the correlations between the various appearances of a domain within a particular query, rather than on the role of the domain in the relations referenced in that query.

The OP/RA expressions are intended to be close to natural language expressions, thereby easier to understand and to formulate than in RA. The query examples have been expressed in a simplified version of ERROL. The utilization of ERROL does not require the learning of just another artificial language, but rather the observation of restrictions enforced upon a language already known to the user. In this way we both propose ERROL as a user oriented interface for OP/RA, and exemplify the capabilities of OP/RA for the semantic definition of natural language oriented query languages. Since both OP/RA and ERROL have a close relationship with natural language sentence combination rules, the use of the OP/RA operators to define the semantics of ERROL is straightforward.
REFERENCES

[1] BILLER, H. On the notion of irreducible relations. In
Database Architecture, Proc. of IFIP Conf., Bracchi, G. and

9-35.

65-98.

[4] CODD, E. F. Extending the database relational model to
capture more meaning. ACM Trans. Database Syst. 4,4 (Dec.
1979), pp. 397-434.

[5] CODD, E. F. Relational databases: a practical foundation for

universal relation assumption and its properties. ACM Trans.
Database Syst. 7,3 (Sep. 1982), pp. 343-360.

[7] MARKOWITZ, V. M. AND RAZ, Y. ERROL: an entity-relationship,
role oriented, query language. In Entity-Relationship Approach
to Software Engineering, Davis, G. C. and all (Eds),


