ERROL: AN ENTITY-RELATIONSHIP, ROLE ORIENTED, QUERY LANGUAGE

by

V.M. Markowitz and Y. Raz

Technical Report #262

February 1983

by

V.M. Markowitz and Y. Raz

Computer Science Department
Technion-Israel Institute of Technology
Haifa, Israel

ABSTRACT

The Entity-Relationship Model is one of the better known semantic data models. Its concepts are based on the way in which people perceive information. Accordingly a query language over the Entity-Relationship Model is expected to have constructs based on the way people communicate, that is, natural language sentences.

The objective of the paper is to propose and define such a query language, called ERROL (Entity-Relationship Role Oriented Query Language). The Entiry-Relationship Model concepts are reviewed and their linguistic aspect is examined.

ERROL is introduced through a large set of characteristic examples illustrating its basic constructs and its syntax is given.

clear boundary between the logical and physical aspects of database management; and communicability, by keeping the model simple enough so that the users could easily understand, use and communicate with one another about the data.

The second objective was only partially fulfilled by the model. The Relational Model allows the user to deal mainly with the representation of things rather than the reality itself, thus hiding much of the semantic structure of the real world.

This places an increased burden on the user, who must try to understand the semantics of the model as well as use it correctly.

Consequently, semantically oriented models have been developed with the Entity-Relationship Model (ERM) [Chen] emerging as the most popular. The ERM reflects a natural view of the world: entities are qualified by their properties and interactions between entities are expressed by relationships which also are qualified by properties. The ERM is thus easy to formulate and to understand. We intend to establish the form of the natural language sentences which fit this view in order to tailor, for the ERM elements, denotations that would suit a query language approaching the natural way of communication.

Concepts underlying the ERM are reviewed in Section 1, in a modified version of the original proposal of [Chen]. How the ERM reflects the surface structure of the enterprise - description sentences is discussed in Section 2. The basic constructs of ERROL are presented in Section 3 with the help of a large set of examples. Section 3 concludes by giving the syntax of ERROL in a BNF like diagrammatic form. Finally, some conclusions are drawn in Section 4.

material or abstract, which can be distinguished from its environment and considered atomic, i.e. not divisible because no part of it will be of interest in that specific environment.

Entities are grouped into entity-sets (e-sets), which have unique names and a type which formalize their time independent aspect. Each e-set is provided with a membership predicate that tests if a given entity belongs to the e-set.

A relationship describes interactions among entities. Relationships are elements of relationship-sets (r-sets), which can be defined as a mathematical relation among n e-sets:

$$R_i = \{[e_1,\dots,e_n] \mid e_j \in E_j, j = 1 \div n\} ,$$

where $[e_1,\dots,e_n]$ represents a relationship.

Not all E_j's have to be distinct, hence their ordering is significant. Two e-sets may be associated by more than one r-set.

A value-set groups together values of a same type (in the same sense as for types of e-sets) and have associated a membership predicate for testing whether a value belongs to it.

A property of an e-set or r-set is defined by a total function from the respective set to a value-set. Requiring the function to be total means that all the elements of some set share the same properties; the existence of an element implies the existence of values for all the properties of the set it belongs to. Note that the property is an association in the sense the r-set is an association. The range of a property, at a given time, is called attribute, having an attribute-name.

An attribute-instance exists, by definition, only when coupled with some entity or relationship. It may be viewed however, as an entity whose identity could independently become of interest. In such a case the attribute should be redefined as an e-set and the corresponding property as a r-set.

A Key of an e-set is a list of properties such that any value from may have any number, [illegible] included, or they be chosen to identify, permanently and uniquely, the entities of the e-set in the system. Their use as identifiers is troublesome because they carry information and are subject to change. Therefore, for every e-set a system-provided value-set of special values called surrogates [Hall] is established as the co-domain of an identifier property; it is called the surrogate-property and the corresponding attribute is called the surrogate-attribute. The surrogate is information-free and protected from the user, who may do no more than cause the system to generate or delete a surrogate. Two surrogates are equal if, and only if, they denote the same entity in the perceived world of entities.

An entity is described by an entity-representation, consisting of the list of all its attribute-instances plus its surrogate. Similarly, a relationship is described by a relationship-representation, consisting of the complete list of all its attribute-instances plus the surrogates of all the participating entities. An entity- or relationship-representation may be viewed as embedding property-representations, each consisting of an attribute-instance plus the common to all surrogate or list of surrogates, of the respective entity or relationship.

Besides the explicitly defined r-sets, there are so called implicit r-sets [Kent], defined by some derivation sequence involving at the initial state explicit r-set, e-sets, or both. In the following discussion r-set shall refer to both explicit and implicit r-sets.

Over a relationship-set may be defined correspondences; we shall restrict the correspondence concept of [San] to the following: given a r-set $R = \{[e_1,\ldots,e_n] \mid e_i \in E_i,\ i = 1 \div n\}$ defined on the e-sets $E_1,\ldots,E_n$, a correspondence, denoted by

$$(E_1,\ldots,E_{k-1},E_{k+1},\ldots,E_n)\ \{E_k\}$$

associates every tuple $[e_1,\ldots,e_{k-1},e_{k+1},\ldots,e_n]$, for which there is some e_k in E_k, such that $[e_1,\ldots,e_{k-1},e_k,e_{k+1},\ldots,e_n]$ is in R, with the set of all the entities e_s of E_k, such that $[e_1,\ldots,e_{k-1},e_s,e_{k+1},\ldots,e_n]$ is in R. An element of $(E_1,\ldots,E_n)$ is called the indexing

They associate the indexing components of a correspondence with the computed values derived from the indexed sets by so-called aggregate functions. Depending on the arity of the i-component, the property would be an entity-property or a relationship-property.

For a given enterprise the complete set of established e-sets, r-sets, properties and attributes, together with the value-sets form the Entity-Relationship Schema (ERS), also known as the enterprise schema.

An ERS may be represented by an Entity-Relationship Diagram (ERD) in the following way:

- e-sets are represented by rectangles labelled with their names;
- r-sets are represented by diamonds labelled with their names;
- attributes are represented by circles labelled with their names;
- properties are represented by arcs connecting the respective sets and attributes;
- the involvement of an e-set in a r-set is represented by an arc connecting their representations.

An Example:

- employees have names, salaries and numbers;
- employees are employed by departments;
- departments are managed by some of the employees;
- departments have numbers and names, and are located on floors;
- suppliers have numbers and names, and have their residence in localities;
- items have numbers, names, colors and types;
- suppliers supply the departments with items, in certain quantities at certain prices;
- suppliers stock items in certain quantities;
- departments request items in certain quantities.

The ERD for the ERS of this example is given in Figure 1 and should be self explanatory.

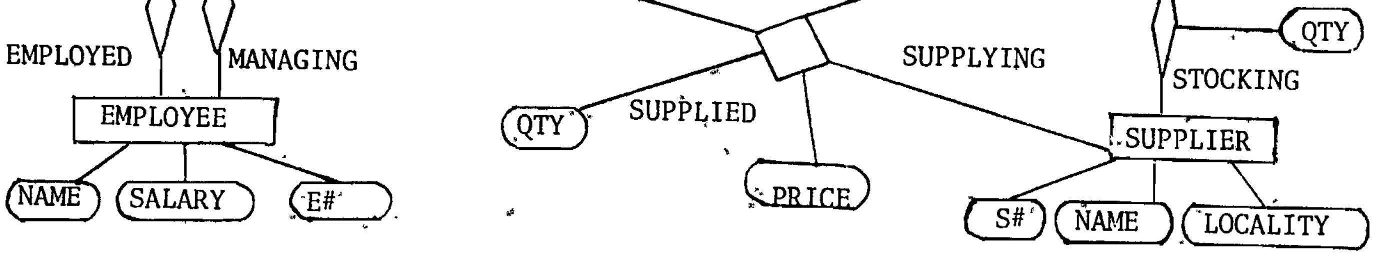


Figure 1 - The ERD and ERS example

We have presented a slightly different and extended set of ERM-concepts, compared to the original proposal of [Chen]. Some of the extensions were adapted from [Pir] and [San]. Other issues concerning the ERM, such as existence dependencies were judged irrelevant to this study and were omitted.

2. THE LINGUISTIC ASPECT OF THE ERM CONCEPTS

The ERM views the enterprise as consisting of entities, having properties, interacting through relationships, which also may have properties. We have no intention of discussing to what extent this view is restrictive or "unnatural". We will try to point out the kind of natural language sentences which fit this view in order to tailor, for the ERS elements, denotations that would suit a manipulation language approaching the natural way of communication, the natural language.

The following ideas were adapted to the ERM from the proposal for a linguistic model for the relational model of [Fur].

Entities are, by definition, atomic in the sense that their decomposition is of no interest in the given enterprise. In the same sense, the attribute-instances are also atomic. Both e-sets and attributes are denoted by names that are, when possible, "real-world" noun names. Properties and r-sets describe associations among elements enterprise, the set of all these sentences form the enterprise sentence description (es-description). Notice that the es-description contains only positive (not formally negated) fact assertions.

We choose to denote an association (r-set or property) by the skeleton (including only the predicate and the object terms) of the participial form of the corresponding es-description sentence. Since a sentence has different paraphrases, accordingly an association will have different denotations. The SUPPLY r-set of Figure 1, for instance, is expressed (the denotations are in uppercase letters) by:

SUPPLIER is SUPPLYING ITEM to DEPARTMENT ;
DEPARTMENT is SUPPLIED with ITEM by SUPPLIER; and
ITEM is SUPPLIED to DEPARTMENT by SUPPLIER.

Notice that 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 denotation of a property belonging to an e-set or r-set is similarly derived. It includes the denotation of the set the property belongs to, and an attribute denotation. The COLOR property of ITEM in Figure 1, for instance, is denoted by:

COLOR OF ITEM; and
ITEM HAVING COLOR

while the QTY property of the SUPPLY r-set in Figure 1 has as one of its denotations: QTY SUPPLIED by SUPPLIER of ITEM to DEPARTMENT. The participation of an e-set or an attribute in a r-set is characterized by the predicate of the paraphrase in which it is in the subject position, called role. It relaxes, in most cases, the ordering-of-e-sets condition, but not as completely as does the "role" of [Chen], where it is defined as asserting the function that the entity performs in the relationship. The roles label in the ERD the arcs representing the participation of the e-sets in r-sets. The ERD thus resembles the surface structure of the es-description sentences. The different r-set denotations provided by the paraphrases of an es-description respond to complex-roles.

3. THE ERROL QUERY LANGUAGE

In this section we present a query language within the Entity-Relationship Model that we call ERROL - Entity Relationship Role Oriented Query Language. ERROL takes advantage of the possibilities posed by the linguistic analogies of the ERM, by using denotations based on the simple sentences of the es-description, and by using constructs similar to the natural language sentence combination. Additionally, ERROL uses correspondence as the basis for set expressions and for formulating derived properties.

The queries in ERROL are intended to be close to natural language expressions, thereby easy to understand and to formulate using few constraints. In this section, the ERROL query language is introduced by a discussion of its basic constructs, and through a set of characteristic examples.

3.1 Basic Concepts

E-set names, attribute names and roles are the ERROL identifiers. We shall denote them as follows:

ENTITY	for e-set names;
ATTRIBUTE	for attribute names;
ROLE	for roles.

The ENTITY, ATTRIBUTE and most of the ROLE identifiers are declared in the ERS over which the queries are expressed. ERROL queries may involve constants (numeric constants, e.g. 2, or strings, e.g. 'ABC') denoted as CONSTANT. ROLEs appear in ERROL prefixed by a single quote (e.g. 'SUPPLIED).

tions of the r-set SUPPLY is:

SUPPLIER 'SUPPLYING ITEM DEPARTMENT.

When there is no ambiguity in a reference to an association, the single quote or a ROLE that has not been declared in ERS may be used. This is the case with most properties. For example, the "SALARY" property may be denoted by:

```
EMPLOYEE 'EARNING  SALARY,  or
EMPLOYEE 'HAVING  SALARY ,  or
EMPLOYEE          'SALARY .
```

Properties of a same r-set may be embedded in a single denotation. For example, the QTY and PRICE properties of SUPPLY may be denoted by:

SUPPLIER 'SUPPLYING ITEM "TO DEPARTMENT "IN QTY "AT PRICE.

If, for some n-arry (n > 2) r-set, interest is in the association of only part of the involved e-sets, a partial denotation may be used:

DEPARTMENT 'SUPPLIED "BY SUPPLIER.

ERROL accepts comments that may be placed anywhere within a query. A comment is prefixed by double quotation marks, is delimitated by spaces, and therefore cannot contain spaces. Comments are useful in making a query look more natural, and thereby self explanatory. In the association - denotations, for instance, the prepositions of the es-description sentences, which are dropped in the denotations, may be included as comments. The above "SUPPLY" denotation could be:

SUPPLIER 'SUPPLYING ITEM "IN QTY "TO DEPARTMENT.

In the following presentation the ERROL key words are underlined. They are used as follows:

GET, TIS, SIT	- in query structuring;
SET	- for set expressions;
AND, OR, NOT	- as logical connectors;
EQ, IN, CONTAINS	- as set operators (possibly preceded by NOT);
NIL	- denotes the empty set;
COUNT,SUM,MIN,MAX	- are aggregate functions.

the brackets,

" (" , ")", " , " , " ; " .

3.2 The GET-CLAUSE

The target elements of a query are stated in the GET-CLAUSE, which is headed by the key-word GET and consists of a list of one or more sublists separated by semicolons. A sublist may contain:

(a) an ENTITY - implying the request for the whole entity-representation, for instance:

GET SUPPLIER;

(b) several ATTRIBUTEs belonging to an ENTITY, possibly followed by the ROLEs corresponding to the respective properties, separated by commas; the last ATTRIBUTE is part of a full property denotation, including the ENTITY - implying the request for a partial entity-representation, for instance:

GET SALARY 'EARNED "BY, NAME 'OF EMPLOYEE;

(c) several ATTRIBUTEs belonging to a r-set, appearing in an embedded property-denotation of properties belonging to a r-set - implying the request for a partial (or whole, if all the ATTRIBUTEs are included) relationship-representation, for instance:

GET QTY 'SUPPLIED ITEM "BY SUPPLIER "TO DEPARTMENT;

Every ENTITY of a sublist (a) or (b), and at least one ENTITY of a sublist (c), will participate in the qualification phrase (T-IS-CLAUSE; see below). If no qualification exists, and there is more than one sublist in the GET-CLAUSE, every sublist is considered as a separate request. The order of the sublists in the GET-CLAUSE, is irrelevant.

The above presentation is partial. A GET-CLAUSE also may include denotations of derived properties and of attributes viewed as e-sets. The definition of the GET-CLAUSE will be completed after these concepts have been introduced.

the target e-sets. This statement is contained in a TIS-CLAUSE (TIS is the abbreviation of "THAT IS"). A TIS-CLAUSE is delimited by the key-words TIS and SIT. It is based on the simple association denotations deducible from the ERS, which are combined in a complex qualification phrase.

3.3.1 Referencing

First of all, we must know how to correlate references to a same e-set. This is done by sufixing the ENTITY with reference variables (REF) which have to start with a "!" sign. For example:

DEPARTMENT!DY, EMPLOYEE!EX .

Such a referencing is less flexible than in the natural language, but is compact and unambiguous. Secondly, the following implicit referencing may be used, when possible, to keep the referencing limited:

(a) a same e-set in the GET-CLAUSE may be correlated by a simple, unreferenced, ENTITY;

(b) an ENTITY suffixed by an "empty" REF (single "!") in the TIS-CLAUSE refers to the corresponding ENTITY of the GET-CLAUSE.

3.3.2 Chaining

Chaining is the simplest way of connecting simple (explicit) associations. A certain denotation of an association reflects how in the corresponding paraphrase of the es-description sentence, an e-set is in the subject position, while all the other e-sets are in the object positions. The last ENTITY of the GET-CLAUSE is the subject in the first association-denotation of the TIS-CLAUSE; therefore it is called the leading subject of the TIS-CLAUSE.

Chaining is based on the natural language sentence combination by relativization. A short connection is assured by the participial position of the following association-denotation. A first query example will illustrate the chaining of two r-set denotations.

(ERROL 1) "Find the names of departments requesting some item stocked by some supplier"

GET NAME "OF DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED "BY SUPPLIER SIT

Restrictions are assertions that are dependent on corresponding property denotations. For instance "LOCALITY = 'PARIS'" is relevant when included in "SUPPLIER 'HAVING LOCALITY = 'PARIS'". Notice, however, that the comparison operators are a kind of ROLE. In the corresponding natural language sentence, the ATTRIBUTE is in a subject position, and the CONSTANT is in an object position.

(ERROL 2) "Find the names of suppliers supplying items of type 1"

GET NAME 'OF SUPPLIER TIS 'SUPPLYING ITEM 'HAVING TYPE = '1' SIT.

A property chaining is realized when the place of the CONSTANT is taken in the restriction by another ATTRIBUTE belonging to some property.

(ERROL 3) "Find the employees working in departments requesting items stocked by suppliers located in the same locations as those supplying red items"

GET EMPLOYEE TIS 'EMPLOYED "IN DEPARTMENT 'REQUESTING ITEM 'STOCKED "BY SUPPLIER 'HAVING LOCALITY = LOCALITY 'OF SUPPLIER 'SUPPLYING ITEM 'HAVING COLOR = 'RED' SIT.

In this example, notice that the different appearances of SUPPLIER do not refer to the same SUPPLIER since no correlation is mentioned.

In the following example, an association which is not explicitly represented in the ERS, has to be "constructively" expressed.

(ERROL 4) "Get the employees that earn more than their manager"

GET EMPLOYEE TIS 'EMPLOYED "BY DEPARTMENT 'MANAGED "BY EMPLOYEE 'HAVING SALARY < SALARY 'OF EMPLOYEE! SIT.

(ERROL 5) "Get the suppliers and the items that are requested by departments that are supplied with these items by these suppliers"

GET SUPPLIER; ITEM TIS 'REQUESTED "BY DEPARTMENT
'SUPPLIED "WITH ITEM! "BY SUPPLIER! SIT.

Observe how it differs from:

(ERROL 6) "Get the suppliers and the items that are requested by departments that are supplied (with items) by these suppliers"

GET SUPPLIER; ITEM TIS 'REQUESTED "BY DEPARTMENT 'SUPPLIED
"WITH ITEM "BY SUPPLIER! SIT .

where "WITH ITEM may be omitted altogether.

The chaining of natural sentences may also be realized by closed relative clause relativization, where the "object-subject" contiguity is broken. The break in the chain is expressed in ERROL by a pseudo-role, indicated by " '* ".

(ERROL 7) "Get the pairs of suppliers and items such that the items are requested by departments to which something is supplied by these suppliers"

GET SUPPLIER; ITEM TIS 'REQUESTED "BY DEPARTMENT!X '*
SUPPLIER! 'SUPPLYING DEPARTMENT!X SIT

The pseudo-role may be used to express cross-product associations in rather unusual, but possible, queries.

(ERROL 8) "Get the pairs of departments and items, such that the items are stocked by some supplier and the department is placed on the 2-nd floor".

GET DEPARTMENT; ITEM TIS 'STOCKED "BY SUPPLIER '*
DEPARTMENT! 'HAVING FLOOR = 2 SIT .

"possible" relationships that are not

"possible" relationships of some r-set R are obtained by the cartesian-product of the e-sets on which the r-set is defined. The complementary r-set of a r-set R is denoted by the negation of the R-denotation.

(ERROL 9) "Get all the items and the departments that do not request them"

GET ITEM; DEPARTMENT TIS NOT 'REQUESTING ITEM! SIT

Similarly, the complementary of a complex (derived) r-set may be denoted by the negation of the chain expressing the r-set.

(ERROL 10) "Get the pairs of suppliers and departments, such that the department is not requesting items stocked by the respective supplier"

GET SUPPLIER; DEPARTMENT TIS NOT 'REQUESTING ITEM
'STOCKED "BY SUPPLIER! SIT

3.3.4 Branching

In addition to chaining, associtions - denotations, whether simple or complex, may be connected by branching. Branching is based on the natural language sentence combination by coordination. The coordination connectors are AND and OR, while the coordination "pivot" is the ENTITY or ATTRIBUTE in the subject position in all the coordinated denotations. In order to enforce a certain order of evaluation brackets may be used.

(ERROL 11) "Get the departments requesting items that are stocked by suppliers from LONDON, and managed by managers earning more than 1000"

GET DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED "BY
SUPPLIER 'HAVING LOCALITY = 'LONDON'
AND 'MANAGED "BY EMPLOYEE 'HAVING SALARY > 1000 SIT

Some binary AND branchings may be reformulated only by chaining.

```
GET QTY 'SUPPLIED "BY SUPPLIER "TO DEPARTMENT "OF ITEM
TIS 'REQUESTED "BY DEPARTMENT! AND 'HAVING COLOR = 'RED'
SIT
```

This may also be expressed as

```
GET QTY 'SUPPLIED ITEM "BY SUPPLER "TO DEPARTMENT TIS
'REQUESTING ITEM! 'HAVING COLOR = 'RED' SIT
```

The OR branching in multi-target queries where more than one sublist are in the GET-CLAUSE, needs references to all the e-sets of the GET-CLAUSE in every branch in order to be unambiguously perceived.

(ERROL 13) "Get the departments and items such that the items are requested by these departments or supplied to these departments"

```
GET DEPARTMENT; ITEM TIS 'REQUESTED "BY DEPARTMENT!
OR 'SUPPLIED "TO DEPARTMENT! SIT
```

(ERROL 14) "Get the departments and items, such that the items are requested by these departments or are red"

```
GET DEPARTMENT; ITEM TIS 'REQUESTED "BY DEPARTMENT!
OR 'HAVING COLOR = 'RED' SIT
```

In the preceding example, the second branch has no references to the DEPARTMENT of the GET-CLAUSE. In such cases the branch is perceived as implicitly completed, through the pseudo-role, with all the unreferenced target e-sets. Thus (ERROL 14) is interpreted as:

```
GET DEPARTMENT; ITEM TIS 'REQUESTED "BY DEPARTMENT!
OR ('HAVING COLOR = 'RED' AND '* DEPARTMENT!) SIT
```

The completion is performed by chaining or AND branching.

element will head an inner TIS-CLAUSE.

(ERROL 15) "Get the departments requesting itmes that are stocked
by suppliers located in LONDON and having type T"
GET DEPARTMENT TIS 'REQUESTING ITEM TIS 'STOCKED ''BY
SUPPLIER 'LOCATED ''IN LOCALITY = 'LONDON' AND 'HAVING
TYPE = 'T' SIT SIT

An inner TIS-CLAUSE contains chains starting with ROLEs corresponding to the pivot subject (e-set in the above example), like the general TIS-CLAUSE. While the general TIS-CLAUSE corresponds to an e-set leading subject, inner TIS-CLAUSEs may qualify attributes or sets as well.

(ERROL 16) "Get the items that are red or blue"
GET ITEM TIS 'HAVING COLOR TIS = 'RED' OR = 'BLUE' SIT SIT

When the pivot is an ATTRIBUTE, the respective TIS-CLAUSE contains chains starting with a comparison operator followed by a single CONSTANT, or with another ATTRIBUTE as the head of a longer chain.

(ERROL 17) "Get the items that are red or are coloured like the items
supplied by LONDON located suppliers"
GET ITEM TIS 'HAVING COLOR TIS = 'RED' OR = COLOR
'OF ITEM 'SUPPLIED ''BY SUPPLIER 'HAVING LOCALITY =
'LONDON' SIT SIT

It has been pointed out that chaining relies on the first and last elements of the association - denotation. When the association is n-arry (n > 2) and the qualification of one of the other elements is required, inner TIS-CLAUSEs are used.

(ERROL 18) "Get the departments requesting items supplied to them by
suppliers located in PARIS, and placed on the 2-nd floor"
GET DEPARTMENT TIS 'REQUESTING ITEM 'SUPPLIED ''BY
SUPPLIER TIS 'HAVING LOCALITY = 'PARIS' SIT ''TO
DEPARTMENT! 'HAVING FLOOR = 2 SIT

Given a r-set, R, defined on the e-sets $E_1,...,E_n$, a correspondence

denoted by

$$(E_1,...,E_{n-1})\ \{E_n\}$$

provides the set of entities from the e-set E_n, associated through R with every tuple $[e_1,...,e_{n-1}]$ (see Section 1). $(E_1,...,E_{n-1})$ is called the i-component, while $\{E_n\}$ is the i-set.

In ERROL the correspondence is denoted by the association implicitly defined by the i-component, followed by the key-word SET and the i-set denotation.

(ERROL 19) "Get the suppliers stocking at least all the items requested by some department"

GET SUPPLIER TIS 'STOCKING SET ITEM CONTAINS SET ITEM 'REQUESTED "BY DEPARTMENT SIT

The above query shows that the i-set may precede the i-component in the denotation, thus integrating naturally the correspondence in a chain. Sets may be compared by set operators (EQ, IN, CONTAINS NOT IN, NOT CONTAINS), with other compatible sets, i.e. sets containing entities of a same type (e-set).

A correspondence denotation omitting the i-component implies a single i-set which refers to a whole e-set.

(ERROL 20) "Get the suppliers stocking all the items"

GET SUPPLIER TIS 'STOCKING SET ITEM EQ SET ITEM SIT

The correspondence is a convenient way of expressing the universal quantifier.

For simple queries, the only effort required from the user is to reformulate expressions involving "all", "at least", "at most", etc., by using sets and set operators. Sometimes the set of interest is not the set of entities belonging to the i-set, but the set of attribute-instances corresponding to a property of the r-set implied by the correspondence. The set-comparisons then involve compatible sets, i.e., containing values from a same value-set.

GET SUPPLIER; DEPARTMENT TIS 'REQUESTING SET QTY
CONTAINS SET QTY 'STOCKED "BY SUPPLIER' SIT

A special key-word, NIL, denotes the empty set. Its use provides an alternative way of expressing some queries involving negation. For example, for

(ERROL 22) "Get the suppliers that do not stock any item",
GET SUPPLIER TIS NOT 'STOCKING ITEM SIT

is equivalent to GET SUPPLIER TIS 'STOCKING SET ITEM EQ NIL SIT

A correspondence denotation may not address all the e-sets on which a r-set is defined. Attention must be paid, in such cases, to the intended meaning. For instance the following two queries clearly differ, although both are based on correspondences with a same i-set and similar (but not identical) i-components:

(ERROL 23) "Get the items that are supplied by all the suppliers"
(i.e. no matter to what department)
GET ITEM TIS 'SUPPLIED "BY SET SUPPLIER EQ SET
SUPPLIER SIT

(ERROL 24) "Get the items that are supplied, to a given department, by all the suppliers"
GET ITEM TIS 'SUPPLIED "TO DEPARTMENT "BY SET SUPPLIER
EQ SET SUPPLIER SIT

An ENTITY from an i-set or an i-component can be qualified.

(ERROL 25) "Get the departments supplied at least by SAXON and PEUGEOT suppliers"
GET DEPARTMENT TIS 'SUPPLIED "BY SET SUPPLIER CONTAINS
SET SUPPLIER TIS 'HAVING NAME TIS = 'SAXON' OR = 'PEUGEOT'
SIT SIT SIT

The entire i-set may also be qualified, provided all the chains in the respective TIS-CLAUSE are headed by set comparison-operators.

```
GET DEPARTMENT TIS 'REQUESTING SET ITEM TIS CONTAINS
SET ITEM TIS 'HAVING COLOR = 'RED' SIT AND CONTAINS
SET ITEM 'STOCKED "BY SUPPLIER 'HAVING S# = '500'
SIT SIT
```

A correspondence may be expressed over a derived r-set. A new ROLE, not declared in the ERS for associations connecting the e-sets referred to by this correspondence, declares the intention of defining a derived r-set. The actual definition is included in a TIS-CLAUSE of the i-set ENTITY.

(ERROL 27) "Get the departments for which all the suppliers, located in LONDON, stock some item requested by them"

```
GET DEPARTMENT TIS 'HAVING SET SUPPLIER TIS 'STOCKING
ITEM 'REQUESTED "BY DEPARTMENT! SIT CONTAINS SET
SUPPLIER TIS 'HAVING LOCALITY = 'LONDON' SIT SIT
```

This is different from:

(ERROL 28)
```
GET DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED "BY SET
SUPPLIER CONTAINS SET SUPPLIER TIS 'HAVING LOCALITY =
'LONDON' SIT SIT
```

which requires "The departments requesting items that are stocked by all the suppliers located in LONDON".

In (ERROL 27) the correspondence is expressed over a derived r-set (the combination of REQUEST and STOCK) while in (ERROL 28) the correspondence is expressed over STOCK.

The derivation may be complex, involving chaining, branching and nesting.

(ERROL 29) "Get the departments for which the items, that are supplied to them or requested by a department placed on the same floor as they are, include all the red items"

```
GET DEPARTMENT TIS 'HAVING SET ITEM TIS 'SUPPLIED "TO
DEPARTMENT! OR 'REQUESTED "BY DEPARTMENT 'HAVING FLOOR
= FLOOR 'OF DEPARTMENT! SIT EQ SET ITEM TIS 'HAVING
COLOR = 'RED' SIT SIT
```

(a) every ENTITY of the i-component has to be referenced at least once; and

(b) all the other references have to be local to this TIS-CLAUSE.

Set expressions may be nested, one inside another, the only restriction is the satisfaction of the above referencing rules.

(ERROL 30) "Get the departments requesting all the items stocked by both, 500 and 701, suppliers"

```
GET DEPARTMENT TIS !REQUESTING SET ITEM EQ SET ITEM TIS
'STOCKED "BY SET SUPPLIER CONTAINS SET SUPPLIER TIS
'HAVING S#='500' OR 'HAVING S#='701' SIT SIT SIT
```

Set nesting is essential for the formulation of more complex universal-quantifier queries. The natural language phrase expressing them has a recursive form, which the ERROL query reproduces. Hence the difficutly of formulating the query in ERROL parallels the difficulty of the corresponding natural language expression.

(ERROL 31) "Get the departments that are supplied by all the suppliers stocking some item, and by every supplier with all the items this supplier stocks"

```
GET DEPARTMENT TIS 'SUPPLIED "BY SET SUPPLIER!Y  TIS
'SUPPLYING DEPARTMENT! "WITH SET ITEM EQ SET ITEM
'STOCKED "BY SUPPLIER!Y SIT EQ SET SUPPLIER TIS
'STOCKING ITEM SIT SIT
```

3.3.7 Derived Properties (Aggregate Functions)

With the help of the correspondence, it is possible to express derived properties. Such a property associates the i-component of a correspondence with values obtained through a computation applied to the i-set of the correspondence. These computations are performed by so-called aggregate functions (af): COUNT, SUM, MAX, MIN. In the correspondence denotation the above key words will replace the key word SET in order to obtain a derived property denotation.

COUNT is applied directly to i-sets of entities and returns the number of entities in these sets. The value-set associated with the

GET ITEM TIS 'STOCKED "BY COUNT SUPPLIER ≥ 3 SIT

Using COUNT might be an alternative to expressions involving negation or set expressions involving the empty set (NIL). Thus the following query is equivalent to (ERROL 22):

GET SUPPLIER TIS 'STOCKING COUNT ITEM = 0 SIT

SUM is applied to sets of property-representations of the property whose denotation appears in the i-set, and processes the attribute-instances of the attribute corresponding to this property. Hence it takes into account any duplicate attribute-instances in the respective set. The value-set associated with the new attribute is the same as the value-set of the attribute corresponding to the above property.

(ERROL 33) "Get the departments requesting a total quantity of items greater than 22000"

GET DEPARTMENT TIS 'REQUESTING SUM QTY > 22000 SIT

MAX and MIN are similar to SUM.

(ERROL 34) "Get the departments whose minimal quantitative requests are no less than 1000"

GET DEPARTMENT TIS 'REQUESTING MIN QTY ≥ 1000 SIT

Derived properties are expressible not only with the help of correspondences over explicit r-sets, but also with correspondences over implicit (derived) r-sets. The denotation and the referencing rules are similar to those for set expressions based on correspondences over derived r-sets.

(ERROL 35) "Get the departments for which there are more than 3 suppliers stocking some item requested by them"

GET DEPARTMENT TIS 'HAVING COUNT SUPPLIER TIS 'STOCKING ITEM 'REQUESTED 'BY DEPARTMENT! SIT > 3 SIT

Notice how this differs from

them"

GET DEPARTMENT TIS 'REQUESTING ITEM 'STOCKED "BY COUNT SUPPLIER > 3 SIT

Similarly, attention must be paid to the derived properties based on correspondences over a special kind of derived r-sets, where the derivation is implied by a partial r-set denotation.

(ERROL 37) "Get the items supplied by more than five suppliers"

GET ITEM TIS 'SUPPLIED "BY COUNT SUPPLIER > 5 SIT

differs from

(ERROL 38) "Get the items supplied to a given department by more than five suppliers"

GET ITEM TIS 'SUPPLIED "TO DEPARTMENT "BY COUNT SUPPLIER > 5 SIT

(ERROL 39) "Get the departments, such that the total quantities, stocked by suppliers supplying these departments, are greater than 10,000"

GET DEPARTMENT TIS 'HAVING SUM QTY 'STOCKED "BY SUPPLIER TIS 'SUPPLYING DEPARTMENT! SIT > 10,000 SIT

In the last query the complete property denotation QTY 'STOCKED ITEM "BY SUPPLIER is implied by the way the SUM is applied, which prevents the elimination of duplicates.

A derived property, precisely as an explicit one, may be chained with other properties, whether they are derived or not.

(ERROL 40) "Get the items requested in a total quantity equal to the quantity in which some supplier stocks them"

GET ITEM TIS 'REQUESTED "IN SUM QTY = QTY 'STOCKED ITEM! "BY SUPPLIER SIT

A "new" attribute may be qualified by a TIS-CLAUSE, the same as an explicit attribute, provided all the chains in the TIS-CLAUSE are of a suitable form (see 3.3.5).

OR ≥ COUNT SUPPLIER TIS 'SUPPLYING ITEM 'HAVING
COLOR = 'RED' SIT SIT SIT .

A derived property may in fact appear in any place in which an explicit one is used, including the GET-CLAUSE, as illustrated by the following examples.

(ERROL 42) "Get the name of the departments together with the number of their employees"

GET NAME 'OF, COUNT EMPLOYEE 'EMPLOYED "BY DEPARTMENT;

(ERROL 43) "Get the sum of the salaries paid in departments, and quantities of stocked items that these departments request"

GET SUM SALARY 'OF EMPLOYEE TIS 'EMPLOYED "BY DEPARTMENT!
SIT, SUM QTY 'STOCKED ITEM TIS 'REQUESTED "BY DEPARTMENT'!
SIT 'OF DEPARTMENT

Every derived-property definition is independent of the rest of the query. A TIS-CLAUSE of such a definition, besides its local referencing, may contain references only to the e-set of the i-component.

3.3.8 Viewing as Entity-Sets

An attribute is dependent on the e-set or r-set it characterizes. Its instances are represented by values from associated value-sets. It might be necessary, however, to refer to an attribute on its own, as we shall show below. In ERROL this is done, simply, by using the attribute as an e-set denotation. The attribute continues to be addressable as formerly, too. Possible ambiguities are resolved by using suitable reference variables (REF) like in the case of e-sets.

(ERROL 44) "Get the localities in which red items are stocked"

GET LOCALITY TIS 'OF SUPPLIER 'STOCKING ITEM 'HAVING
COLOR = 'RED' SIT

The answer will be a set of LOCALITY-instances, not a set of property-representation corresponding to LOCALITY.

appears with type T in some other item"

GET COLOR!C; TYPE!T TIS 'OF ITEM 'HAVING COLOR!C 'OF ITEM
'HAVING TYPE ≠ 'T' AND 'OF ITEM 'HAVING COLOR = 'RED' SIT

Notice that the role used for the attribute viewed as e-set is freely chosen, since the only connection of the attribute is with the e-set it is characterizing.

3.3.9 Factorizing Common Chain-Suffixes

Successive chains of a TIS-CLAUSE might have a common suffix (tail). It would be more convenient to express it only once, particularly when it is complex.

In ERROL, the common tail of successive chains is factorized out using commas, according to the following rules:

(a) a comma preceding a coordination connector (AND, OR) marks the place of the common tail;

(b) a comma preceding an ENTITY or an ATTRIBUTE marks the beginning of the common tail.

(ERROL 46) "Get the items requested by, and supplied to, departments placed on the second floor"

GET ITEM TIS 'REQUESTED "BY, AND 'SUPPLIED "TO, DEPARTMENT
'HAVING FLOOR = 2 SIT

which is equivalent to

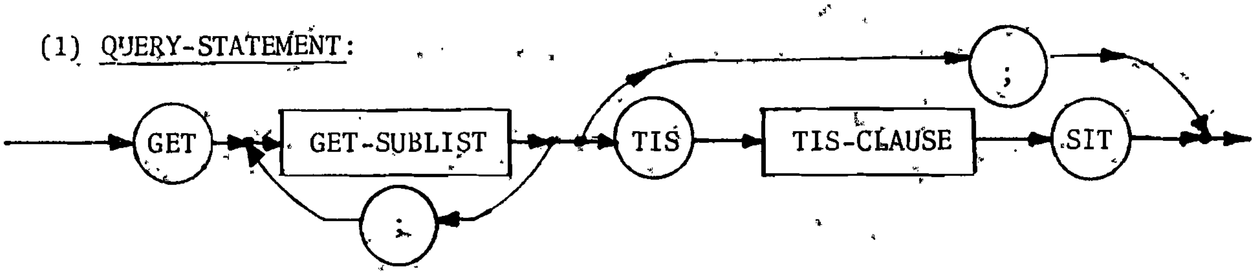
GET ITEM TIS 'REQUESTED "BY DEPARTMENT 'HAVING FLOOR = 2
AND 'SUPPLIED "TO DEPARTMENT 'HAVING FLOOR = 2 SIT

Note that a common tail does not mean the addressing of the same ENTITY. In the above query, for instance, the items are not necessarily requested by, and supplied to, the same department. That is why the following query differs from the former one:

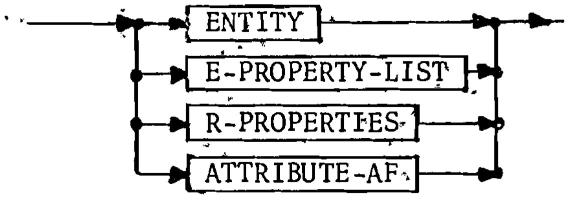
(ERROL 47) GET ITEM TIS 'REQUESTED "BY DEPARTMENT TIS 'SUPPLIED "WITH
ITEM! AND 'HAVING FLOOR = 2 SIT

The following is a slightly simplified diagrammatic BNF syntax for ERROL. Lower-case letters are used for representing terminal strings that are dependent on the ERS or on the user.

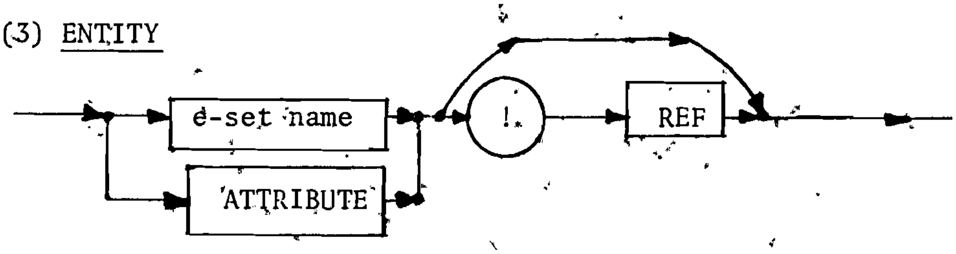
(1) QUERY-STATEMENT:



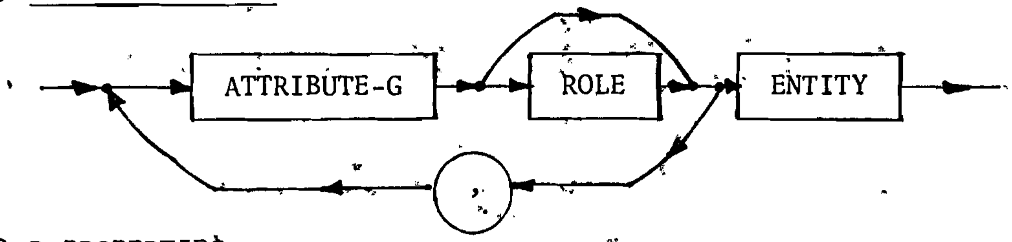
(2) GET-SUBLIST:



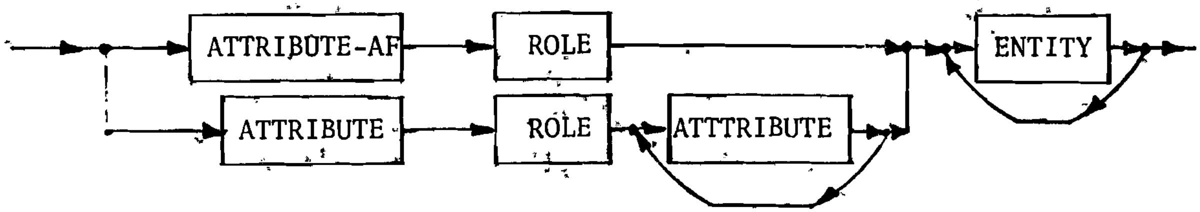
(3) ENTITY



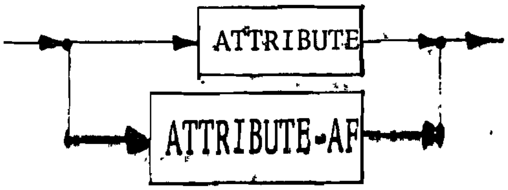
(4) E-PROPERTY-LIST:



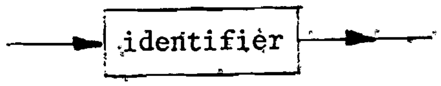
(5) R-PROPERTIES



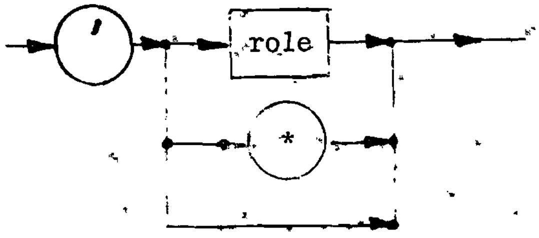
(6) ATTRIBUTE-G



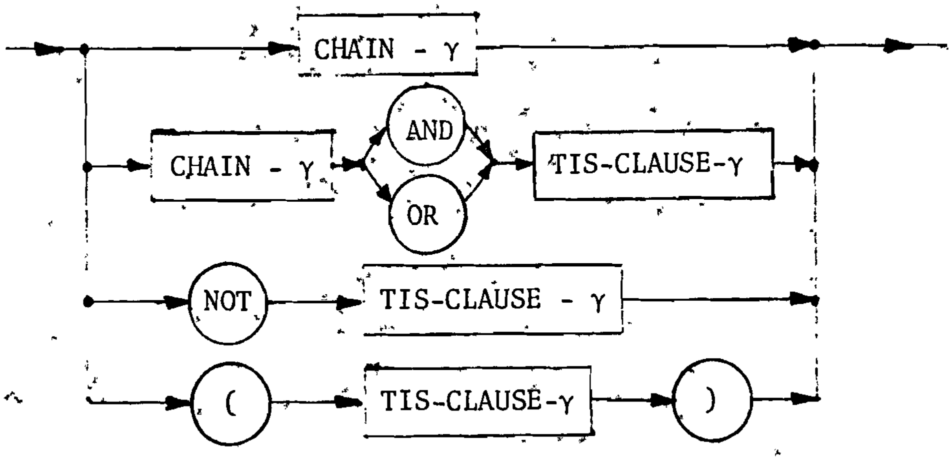
(8) REF:



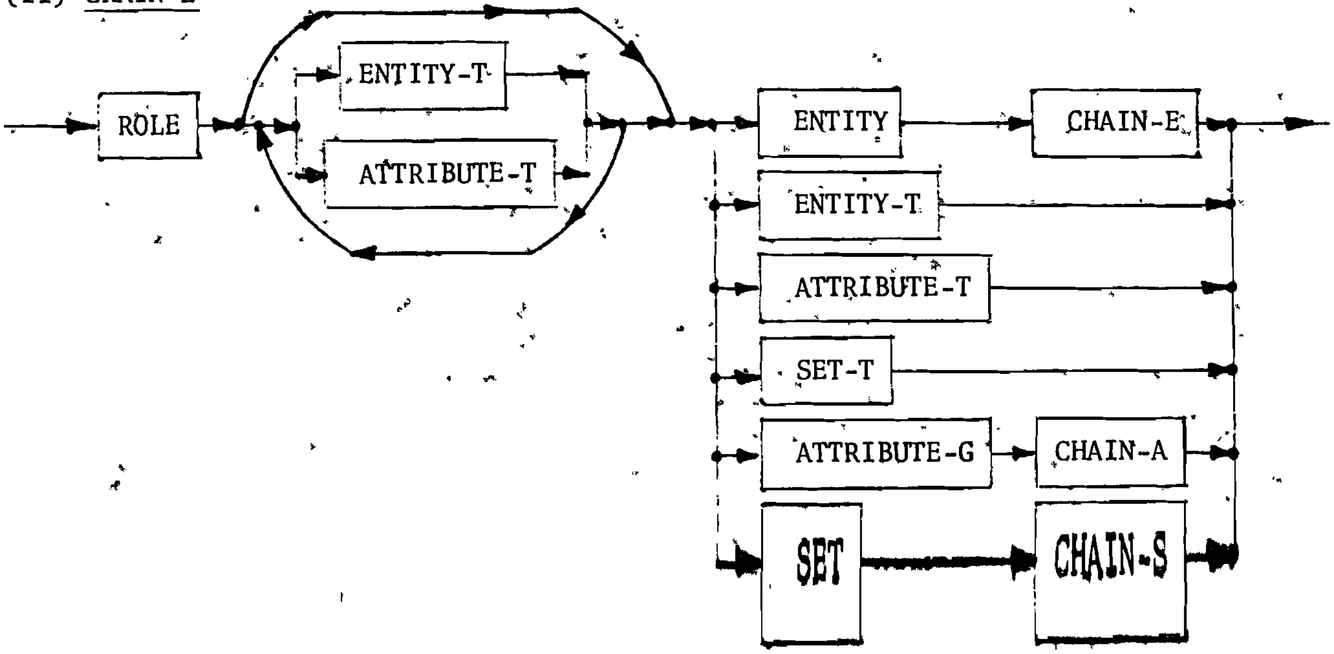
(9) ROLE:

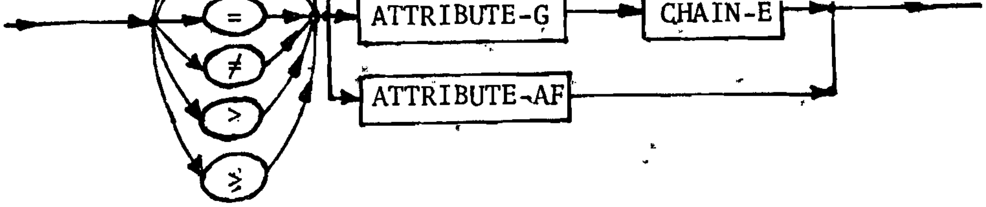


(10) TIS-CLAUSE - γ (γ = E XOR A XOR S):

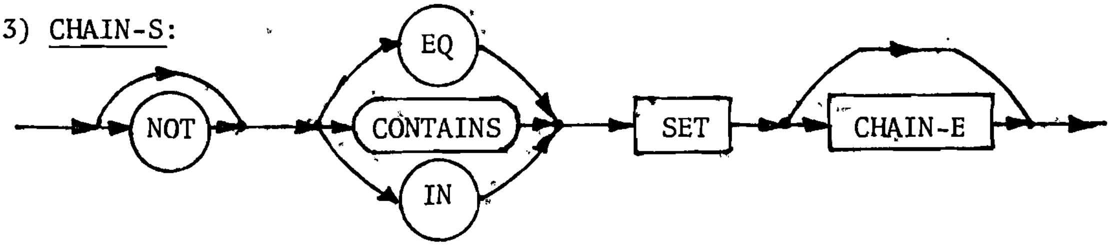


(11) CHAIN-E

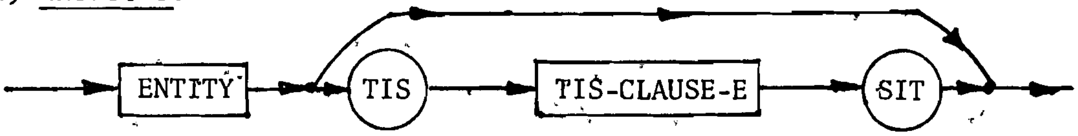




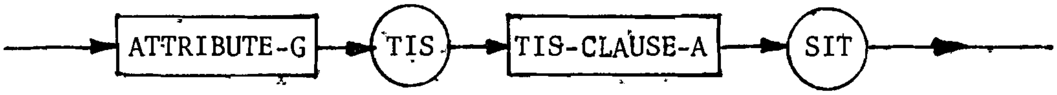
(13) CHAIN-S:



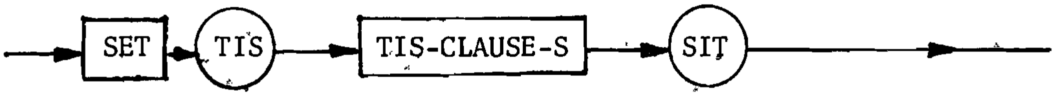
(14) ENTITY-T:



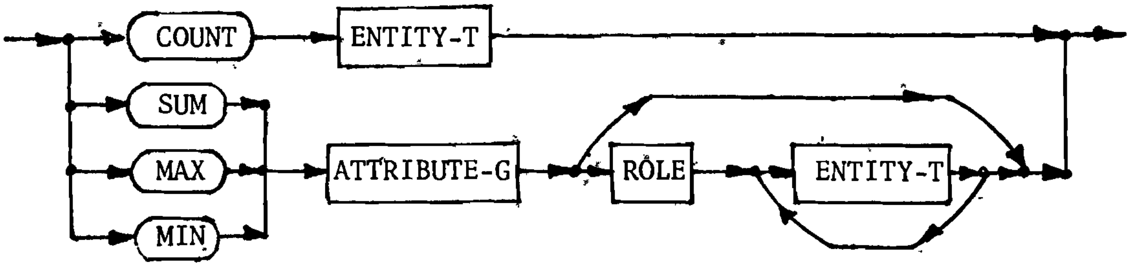
(15) ATTRIBUTE-T:



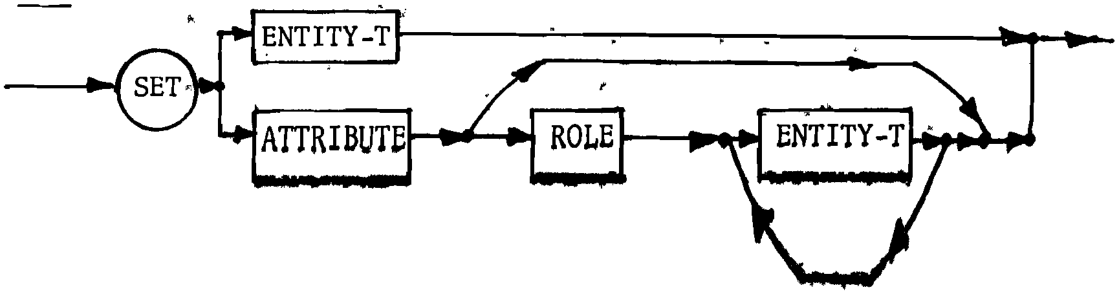
(16) SET-T:

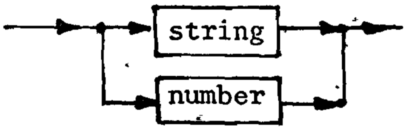


(17) ATTRIBUTE-AF

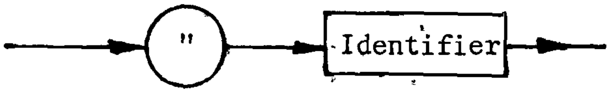


(18) SET:





(20) COMMENT:



Syntax Rules Not Included in the BNF Definition

(a) The factorization of common chain-suffixes (Section 3.3.9);

(b) The referencing rules for set expressions based on correspondences over derived r-sets (Section 3.3.6).

4. CONCLUSION

This paper has proposed ERROL - a query language over the ERM. ERROL draws on analogies with natural language constructs, based on the linguistic aspect of the ERM concepts. As such, ERROL is structurally different from all the query languages based on the ERM that have appeared in the literature. Moreover, according to the definition we know, all the languages (Executable Language [Atz], CLEAR [Poo], CABLE [Sho]) have less expressive power than ERROL; GORDAS [Elm] is the exception.

We tried not to depart from the analogies to the natural language. However, our goal was to make a construct simple and unambiguous, rather than natural. For instance, the referencing in ERROL does not resemble the natural language ways of correlation. The use of sets also is more restrictive, or less natural, than that of the natural languages quantifiers. However, we believe ERROL still remains close to the natural perceiving of information and the natural way of communication. If our assumptions are correct, ERROL should be easy to comprehend, learn, and master, even for casual users.

The ERS describes the enterprise structure. At the data representation level, the entities are represented by the entity-representations and the relashionships are represented by relationship-representations. All are lists of attribute-instances, some of them being surrogates.

We choose to organize these data in relations, which are simple structures and benefit from being the basis of an advanced data model, the Relational Model (RM). Since a link between ERM and RM is established, by choosing the relation to be the structural unit of the ERM data-representation level, the ERM could benefit by somehow inheriting the Relational Algebra (RA) of the RM [Codd]. The RA operators cannot be used without modification in an ERM environment, however, without endangering the semantic structure of the ERM. In [MR] a set of semantic operators (s-operators) is proposed to form the manipulative part of the ERM, by adapting the RA operators to the structural constraints of the ERM. The linguistic analogies of the s-operators are investigated and relativization, coordination, referencing and quantification, which are the basis of the ERROL constructs, are shown to be easily expressible with their help. Accordingly, the description of the semantics of the ERROL basic constructs using the s-operators is straightforward [Mar].

BIBLIOGRAPHY

[Atz] P. Atzeni, P.P. Chen - "Completeness of Query Languages for the Entity-Relationship Model", E-R Approach to Information Modelling, P.P. Chen ed., ER Institute, 1981, pp. 111-124.

[Chen] P.P. Chen - "The Entity-Relationship Model - Toward a Unified View of Data", TODS, March 1976, pp. 9-35.

[Codd] E.F. Codd - "Relational Databases: A Practical Foundation for Productivity", CACM, Feb. 1982, pp. 109-117.

[Elm] R. Elmasri, G. Wiederhold - "GORDAS: A Formal High Level Query Language for the E-R Model", same as [Atz], pp. 49-72

[Fur] A.L. Furtado - "Formal Aspects of the Relational Model", Information Systems, 1978, pp. 131-140.

[Hall] P. Hall, J. Owlett, S. Todd - "Relations and Entities", IFIP Conference on Modelling in Database Systems, 1976, North Holland, pp. 201-220.

[Kent] W. Kent - Data and Reality, North Holland, 1978.

[Pir] A. Pirotte - "The Entity-Association Model", Int. Computing Symposium, 1977, pp. 581-597.

[Poo] G. Poonen - "CLEAR: A Conceptual Language for Entities and Relationships", Proc. ICMOD 1978, pp. 95-113.

[San] C.S. dos Santos, E.J. Neuhold, A.L. Furtado - "A Data Type Approach to the E-R Model", E-R Approach to System Analysis, North-Holland, 1980, pp. 103-119.

[Sho] A. Shoshani - "CABLE: A Language Based on the Entity-Relationship Model", (as presented in "Data Models", D. Tsichritzis, F. Lochovsky, eds., Prentice Hall, 1982).

[Mar] V.M. Markowitz - "ERROL - An Entity Relationship Query Language", M.Sc. thesis, Technion, January 1983.

[MR] V.M. Markowitz, Y. Raz - "Modifying the Relational Algebra for Use in an Entity-Relationship Environment", TR#263, Technion, January 1983.