THE SAGE* GEOGRAPHIC ANALYSIS SYSTEM

R.CAUBET A.HAMEURLAIN
Conference Master Assistant
Université Paul Sabatier - UER MIG-LSI**
118 route de Narbonne - 31062 TOULOUSE CEDEX
FRANCE

ABSTRACT

This paper describes the design and implementation of a geographic analysis system, SAGE. We present a classification of geographic analysis database queries and two geographic data manipulation modes.

Then, after establishing a database model for geographic objects, we specify of integrity constraints on the time/space elements of the geographic database and discuss a means of ensuring the structural coherence of geographic objects.

KEYWORDS

databse, relational language, geographic objects, cadaster, graphics, integrity constraints, structural coherence.

1. INTRODUCTION

A particular geographic zone at all times can be described by the series of land related transactions carried out within the zone. These transaction, along with information such as land utilisation, resource availability, occupying social groups etc... can be used to determine a zone's evolution in time and space and hence can be helpful in the general administrating and planning of geographical zones. SAGE is a system which helps this zone administration and planning.

A cadaster (or land registry) being and important source of dynamic type data, is quite difficult to manipulate manually (CHE80). We recall that a cadaster is composed essentially of a map indicating land parcel layouts.

In an effort to render more manageable the manipulation of a region's cadaster, we have designed and implemented a specialized graphic interface for use between a geographic database and users. This graphic interface will serve as an interactive tool for the creation of geographic objects and will permit the user to view, through the 2D and 3D screen representation of these objects, the effect of particular geographic object manipulations.

2. USER REQUIREMENTS IN GEOGRAPHIC/CADASTRAL ANALYSIS

Our goal is to provide users with simple, effective and appropriate operators for realizing their applications, tools capable of satisfying the various queries users may have.

These various queries for example may range :

- from requesting the parcel layout of a particular neighberhood, requesting all recorded transactions between two landouness, or requesting the parcelling history of a given goegraphic zone (ADI81) (ADI85).
- to joining the land parcels belonging to a particular landouner, or expanding a land parcel to a new land boundary (i.e.register the sale of a land parcel).
- to determining whether or not a particular manufacturing plant is in a particular province or if a particular province is in a particular region (HAM84).

In an effort to satisfy these varied user queries, we have designed and implemented more specifically a graphic interface enabling a user to construct and manipulate various geographic objects represented in a geographic database.

The realization of this interface required us to answer the important questions : How is geographical data to be gathered or input to the database ? How is geographical data to be structured, presented and manipulated ?

This geographic data, we have identified may be either factual data describing for example, layout (or geometric shape) of a land parcel.

Graphic data on geographic zones, incidentally, is obtained from the digitalization of geographic zone photographs taken by land sur-

---

* French acronym for "Système d'Analyse Géographique"
** Laboratory "Langages et Systèmes Informatiques"

veying satellites such as SPOT, LANDSAT... This digitalization in conjunction with certain image processing techniques is capable of yielding, among other things, the contours of land parcels, zones, rivers,or roads (YAN80) (CHA80). These contours can be represented by geometric shapes, namely by polygons, lines and points, hence the graphic data.

3. CLASSIFICATION OF DATABASE QUERIES

Our classification of database queries is based essentially on the class (factuel,graphic) of the data requested in the database.

We define two classes of user queries :
1) linear queries
2) compound queries

We do not elaborate in the following discussion queries operating strictly on factual data corresponding to classical database (DB) queries.

3.1. Linear queries

A database query Q is linear if it is executed in one of the following orders :

a) Qa1 : in Factual DB : obtain factual data satisfying specified criteria

Qa2 : in Graphic DB : obtain graphic data corresponding to factual data obtained in Qa1

Qa3 : geometric operator is executed on graphic data

b) Qb1 : in Graphic DB : select geometric shapes displayed on the screen

Qb2 : in Factual DB : obtain factual data corresponding to shapes identified in Qb1

c) Qc1 : in Graphic DB : select geometric shapes displayed on the screen

Qc2 : in Graphic DB : obtain graphic data corresponding to shapes identified in Qc1

Qc3 : execute geometric operator on graphic data.

Thus, there are three types of linear queries.

a) Factual Graphic Queries

These queries operate first on factual data then on graphic data and are carried out in the a order. For example the factual graphic query "Fuse the polygons belonging to X" is carried out in the Qa1.Qa2.Qa3 order where during Qa1 the obtains the numbers identigying the polygons requested, during Qa2 the system obtains graphic data corresponding to the identified polygons, and during Qa3, the system applies the FUSION operator (which for example may be a binary operator) on the selected graphic data.

b) Graphic Factual Queries

These queries operate first on graphic data then on factual data, involve no use of geometric operators and are carried out in the b order.For example, the Graphic Factual query "what is the date of creation of the polygon selected on the screen ?" is carried out in the Qb1.Qb2 order where during Qb1 the system determines the identifier of the polygon selected on the screen, and during Qb2 the system obtains the requested date of creation corresponding to this polygon identifier.

c) Graphic Queries

These queries operate strictly on graphic data and are carried out in the c order. The following are examples of Graphic Queries :

. "Determine the center of the polygon selected on the screen"

. "Calculate the area and perimeter of the polygon selected on the screen"

. "Fuse the polygons selected on the screen"

. "Expand the polygon selected on the screen" (specifying the new boundaries")

3.2. Compound queries

A data query Q is compound if it is executed in one of the following manners :

$$1.\ (\ \text{Qa1} \left\{\begin{matrix}\text{AND}\\ \text{OR}\\ \text{EXCEPT}\end{matrix}\right\} \text{Qa1}\ )\ \text{Qa2}\ .\ \text{Qa3}$$

$$2.\ (\ \text{Qb1} \left\{\begin{matrix}\text{AND}\\ \text{OR}\\ \text{EXCEPT}\end{matrix}\right\} \text{Qa1}\ )\ \text{Qa2}\quad \text{Qa3}$$

where AND, OR and EXCEPT represent respectively set intersection, set union and set difference, and Qa1 , Qa2 , Qb1 and Qb2 are as defined priviously.

These queries permit the user through use of AND/OR/EXCEPT operators to link factual queries to graphic queries as would be required in the compound query "Fuse the polygons selected on the screen if they belong to X".

This query would be carried out, for example, in the (Qb1 AND Qa1) Qa2.Qa3 order (order 2).

Where during Qb1 the system determines the identifiers of the specified polygons (yielding set 1), during Qa1 the system determines the identifiers of all polygons belonging to X (yielding set2), and during Qa2 the system obtains graphic data corresponding to the polygons included in set 1 AND (intersection) set 2, and where during Qa3 the goemetric operator FUSION is executed.

Note that if the intersection of set 1 and set 2 was empty, Qa2 and Qa3 would not be executed. To ensure complete execution of a compound query, the system under certain conditions can choose a particular evaluation order (order 1 or 2) to be followed.

The following summarizes our classification of database queries :

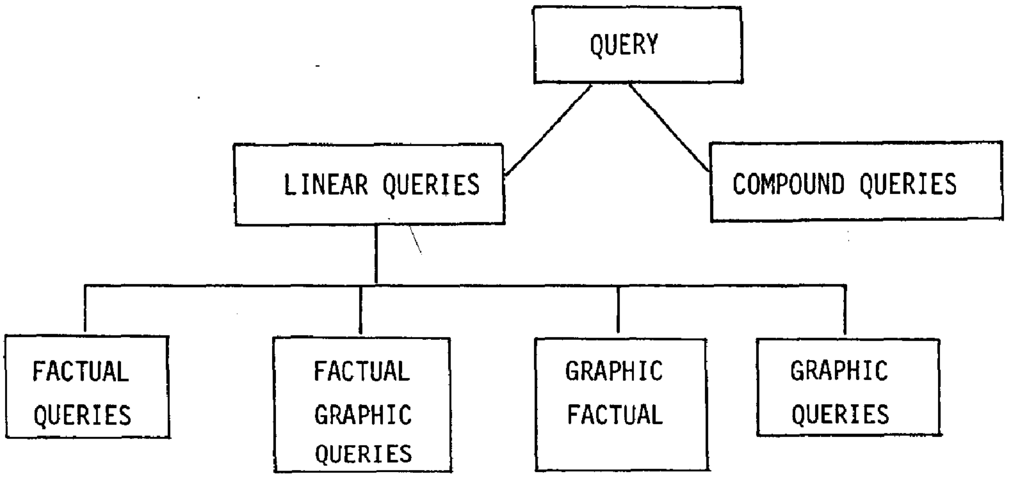


Fig.1 : Classification of queries in a geometric (factual and graphic) database

4. GEOGRAPHICAL DATA MANIPULATION MODES

Relational languages such as SEQUEL (AST75) of System-R (AST76) QUEL (HEL75) of the INGRES system (STO76) and QBE (ZLO75) (ZLO81) proposed for use in relational models (COD70) have been defined and implemented for databases containing factual data. The languages do not provide the basis for manipulating graphic data. Recently however, several other relational languages have been extended to permit the manipulation of graphic data (e.g. QPE of IMAID system (CHA80), GRAIN (CHA77), ELF (YAM80)).

We in turn, for the handling of graphic data, propose two modes of manipulation :

1. a graphic mode, in which users are offered, via the specialized graphic interface, sophiscitated graphic tools for manipulating graphic data.
2. an alphanumeric mode in which users are offered an extended relational query language for manipulating graphic data.

4.1. The graphic interface

The graphic interface enables the user

- to specify which geometric objects are to be included in certain land transactions
- to carry out land transactions (i.e. execute geometric operators on factual and graphic data in the database)
- to invoke certain graphic manipulation aids.

We have implemented a set of geometric operators for use in certain geographical/cadasteral analyses (cf.Appendix 2) ; the geometric shapes are drawn by these operators using the T.C.S./G.K.S. graphic operator library.

The implementation required that we

- divide tho graphic screen into work, dialog, command zones
- provide for graphic data input and update of factual data
- enable specification of geometric objects by the user
- provide for graphic data display (in color and in varied textures).

The following example illustrates how the user creates geometric objects (e.g. geographic zones, land parcels, rivers, roads, etc...)from objects already recorded in the database.

These existing objects were created using the TRANS operator (cf.Appendix 2) and graphic entry tools (THA83).

Example : "Expand the specified land parcel to a specified new boundary".

The query is executed in two steps :

1. The system determines the identifier of the specified land parcel and gets the corresponding graphic data.
2. The user indicates on the graphic screen the various new boundary points to which the specified land parcel must expand.

4.2. Query language extension for geographic data manipulation

As already mentioned, relational languages need to be extended in order to satisfactorily handle geographic objects. The target language must make it possible to user existing relationships between factual and graphic data and,since the operands used by geometric objects (as mentioned in 4.1) must be able to construct set operands by executing queries in the base relational language.

The proposed extension consists of adding to the relational language the set of goemetric operators (cf.Appendix 2).

Extended language syntax

The following is the BNF representation of the extended language for geometric object manipulation :

```
             <QUERY> ::= <GEOM_OPERATOR> <FACTUAL_DB_QUERY>
     <GEOM_OPERATOR> ::= <CONST_OPERATOR>/<AGREGAT_OPERATOR>/
                         <CONVERSION_OPERATOR>/<LINK_OPERATOR>
  <FACTUAL_DB_QUERY> ::= <BNF representation of a data manipulation language>
    <CONST_OPERATOR> ::= FUSION/EXPAND/INTER
  <AGREGAT_OPERATOR> ::= AREA/PERIMETER/LENGHT/CENTER
<CONVERSION_OPERATOR> ::= TRANS VIEW
     <LINK_OPERATOR> ::= INCLUSION_SP/INCLUSION_LP/INCLUSION_PP
```

Thus, a user is capable with this language to properly and effectively manipulate geometric objects.

Example : "Fuse the land parcels belonging to X"

This query illustrates well the relationship between the factual and graphic objects of the database.

5. THE CADASTERAL INTEGRITY SYSTEM

5.1. The relational database schema(cf.Appendix3)

Given that each geometric object is represented in the database using two types of information, factual and graphic data, the relational schema of the cadasteral application (HAM85) becomes :

GRAPHIC RELATION (LINESGNUM, OBJNUM, X1,Y1,X2,Y2)

FACTUAL RELATION (OBJNUM, NAME, ORIGDATE , EXPDATE..)

Note that geometric objects are all formed of linesegments (CHA80) , (BRA80) and (BEC80).

As the modelled geographic objects are mostly complex geometric objects and ordering of the associated graphic relation tuples had to be ensured (hence LINESEGNUM in the GRAPHIC-RELATION) as well as the ability to identify the complex object considered in the GRAPHIC-RELATION (hence the OBJNUM attribute in the GRAPHIC-RELATION).

5.2. Integrity constraints

Since the modelled objects as mentioned earlier are rather complex, it is important to introduce a series of integrity constraints ensuring the coherence of an objects corresponding factual data and to introduce controls on the structural coherence of an object's structure (MEI80) or graphic data.

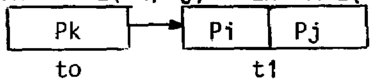
5.2.1. Time related integrity rules

R1 : the date of origin of a land parcel, or region ... must be inferior to its expiry date

$\forall i \in [1,k]_t$ ORIGDATE(Pi) < EXPDATE(Pi)

R2 : the date of origin of a land parcel or region ... must equal the expiry date of the corresponding parent parcel

ORIGDATE(Pi/Pj) = EXPDATE(Pk)



R3 : all the land parcels, regions ... originating from one common parent parcel during an EXPAND must have the same data of origin

if EXPAND(Pk,c1,c2...cm) = $\{Pi\}$

then $\forall i \in [1,N]$ ORIGDATE(Pi) = t

R4 : all land parcels, regions ... fused at the same time (into Pk) must have the same expiry date

if FUSION(Pi,Pj) = Pk

then EXPDATE(Pi) = EXPDATE(Pj)

5.2.2. Space related integrity rules

R5 : Global map area preservation

$$\forall t \sum_{i=1}^{w(t)} \text{AREA}(Pi) = k \pm \text{EPS}$$

k = original map area
EPS = max allowable error
w(t)= number of land parcels,regions at time t

R6 : Map area preservation after manipulations

a) when a set X of land parcels,regions... X($|X| \geqslant 2$) is fused together into a land parcel (Pj) or region ..., the area of Pj must equal the sum of the areas of the parcels (Pi), regions ... of X.

At time t let X(t) = Pi/$|i| \geqslant 2$ i=1,2,...N

if FUSION(X(t)) = Pj ($j \neq i$)

then $\text{AREA}(Pj) = \sum_{i=1}^{N} \text{AREA}(Pj) \pm \text{EPS}$

b) when a land parcel (Pj), region ... is expanded into a set X of land parcels (Pj) or region ..., the area of Pj must equal the sum of the areas of the parcels (Pi), regions... of X.

$\underline{\text{if}}$ $\text{EXPAND}(Pj, c1, c2 \ldots cm) = Pi \qquad i=1,\ldots N$

$$\underline{\text{then}}\ \text{AREA}(Pj) = \sum_{i=1}^{N} \text{AREA}(Pi) \pm \text{EPS}$$

5.2.3. Structural coherence of goemetric objects

Before carrying out a geometric construction operation (i.e. FUSION, EXPAND) pre and post conditions related to the structural coherence of the geometric objects involved must be verified.

The postconditions are actually updates of the factual and graphic parts of the database ; the preconditions are object coherence tests.

Here, we present only the EXPAND operator preconditions :

Let S1,...,SN be the linesegments composing the contour of a polygon $Pk = S1 * S2 * \ldots * SN \quad (N \geqslant 3)$

Let S1',...,SN' be the line segments composing the new boundary B between two polygons obtained from a EXPAND on Pk

$$B = S1' * S2' * \ldots * SM' \qquad (M \geqslant 1)$$

EXPAND preconditions (EP) can be expressed as follows :

$\text{EP1} : \forall i\ [2,M]\quad \text{INIT-EXTREM}\ (Si')_B \in \text{ON}(Pk)$

$\qquad \text{EXTREMITES}\ (B)\quad \text{CONTOUR}\ (Pk)$

$\qquad \forall i \in [1,M]\quad \text{MIDDLE}\ (Si')_B \in \text{INSIDE}\ (Pk)$

$\text{EP2} : \forall i \in [1,M]\ \forall j \in [1,N]\quad YSi' \cap YSj = \emptyset$

$\text{EP3} : \forall i \in [1,M-1]\ \forall j \in [i+1,M]\quad YSi' \cap YSi' = \emptyset$

$\qquad \forall i \in [2,M]\quad \text{ADJACENT}\ (\text{INIT-EXTREM}(Si')_B) = 2$

Fig.2 illustrates various EXPAND configurations in which these preconditions are met

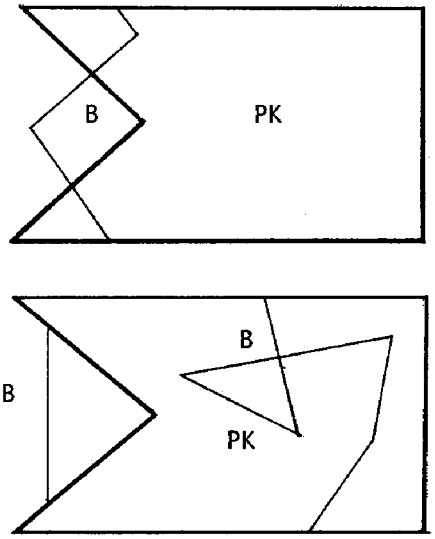


Fig.2 : EXAMPLES of verified EXPAND preconditions

6. CONCLUSION

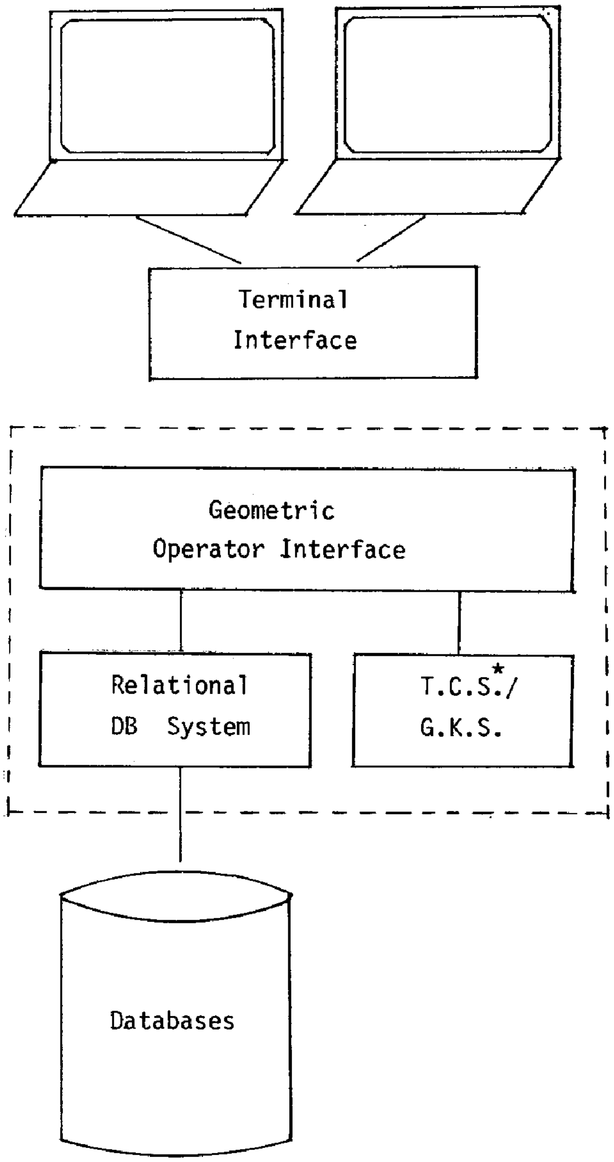
In this paper, we have presented the various cadasteral object handling capabilities of SAGE and have shown how relational languages can be extended to effectively manipulate objects recorded in a cadasteral database.

Our study has shown how a relational database system can be used to describe complex geometric objects and how it can be used to implement geometric operators.

We project to fully implement both the presented graphic interface and an integrity system taking into account the dynamic aspects of the geometric objects in the cadasteral database.

APPENDIX

A1. The SAGE Architecture



* Terminal Control System (Tektronix)

A2. The SAGE geometric operators

(In the following let F be a set of polygons
let L be a set of linesegments,
let P be a set of points).

OPERATOR	DEFINITION	SYNTAX	RESULT
FUSION	Forms a new polygon by eliminating line segments common to two polygons	FUSION (Fi,Fj)	$Fk \in F$ Fk is one polygon
EXPAND	Forms two polygons sharing a common boundary from one polygon	EXPAND (Fk,P1,P2...Pn)	$Fi,Fj \in F$ $i \neq j$
INTER	Determines the segments of a line lying within a polygon	INTER (Li,Fi)	Lj , a sequence of linesegments
AREA	Calculates the AREA of a polygon	AREA (Fi)	S, a scalar
PERIMETER	Calculates the perimeter of a polygon	PERIMETER (Fi)	S, a scalar
LENGTH	Calculates the length of a line	LENGTH (Li)	S, a scalar
CENTER	Calculates the center of a polygon	CENTER (Fi)	$Pi \in P$
DISTANCE	Calculates distance between two points	DISTANCE (Pi,Pj)	S, a scalar
TRANS	Converts a squelettal image into a relation	TRANS (pt1,pt2,...,ptn)	a relation
VIEW	Displays a relation (or part of) as a squelettal image	VIEW (Relation, C1)	a squelettal image
INCLUSION_PL	Checks whether or not a point is on a line	INCLUSION(Pi,Lj)	TRUE or FALSE
INCLUSION_PF	Checks whether or not a point is in a polygon	INCLUSION(Pi,Fj)	TRUE or FALSE
INCLUSION_LF	Checks whether or not a line is in a polygon	INCLUSION(Li,Fj)	TRUE or FALSE
SHIFT	Shifts a polygon in a particular direction D	SHIFT(Fi,D)	Fj, a shifted polygon
TRANSCALE	Transforms the scale of a polygon by multiplying its vertices coordinates by a factor V	TRANSCALE (Fi,V)	Fk, a scaled polygon

A3. Relational schema of cadastral application

%PARCEL%

GRAPHIC_PARCEL(LINESGNUM,PARCELNUM,X1,Y1, X2,Y2);

FACTUAL_PARCEL(PARCELNUM,NAME,ORIGDATE, EXPDATE,COLOR,TEXTURE);

FILIATION_PARCEL(FINALPARCELNUM, INITIALPARCELNUM);

MUTATION(PARCELNUM,VANDACCOUNTNUM, BUYACCOUNTNUM,MUTDATE, POSSESSION_PERIOD);

INDIVIDU(ID,NAME,ADDRESS,PROFESSION, ACCOUNT_NUM);

%ROAD%

GRAPHIC_ROAD(LINESGNUM,ROADNUM,X1,Y1,X2,Y2);

FACTUAL_ROAD(ROADNUM,NAME,COLOR);

%RIVER%

GRAPHIC_RIVER(LINESGNUM,RIVERNUM,X1,Y1,X2,Y2);

FACTUAL-RIVER(RIVERNUM,NAME,COLOR);

%LOCATION of city, a manufactury, and so on, as a point object%

LOCATION(OBJNUM,X1,Y1);

BIBLIOGRAPHY

ADIDA, M. ; La gestion du temps dans les SGBD, Bases de données - Nouvelles perspectives. Rapport du groupe BD3, INRIA-ADI : 145-148, 1983.

ADIBA, M. ; Notion de temps dans les bases de données généralisées, Modèle et Base de Données, 1 : 3-11, 1985.

ASTRAHAN, M.M. ; CHAMBERLIN, D.D. Implementation of Structured of English Query Language, CACM, 18(10) : 580-588, 1975.

BRATBERGSENGEN, K. ; LARSEN, R. Design of a relational database system, RUNIT Report, 11/80, Univ.of Trondheim (Norway), 1980.

BECERIL, J.L. ; CASAJUANA, R. ; LORIE, R.A. GSYSR : a relational database interface for graphics, Lecture Notes in Computer Sciences, 81 : 459-474, 1980.

CHANG, N.S. ; FU, S.K. A query language for relational image database systems, Proc.of the IEEE WPDDM*, 68-73, 1980.

CHEYLAN, J.P. ; DESBORDE, F. Banque de données et informations administratives pour une observation continue de la formation des espaces péri-urbains, Rapport LISH/T19,1982.

CODD, E.F. ; A relational model for large shared data banks, CACM, 13(6) : 377-387,1970.

HAMEURLAIN, A. ; Les opérateurs géométriques. Conception et réalisation, Thèse de 3ème Cycle, Univ.Paul Sabatier (Toulouse), 1984.

HAMEURLAIN, A. ; BAZEX, P. ; CAUBET, R. La manipulation de figures géométriques dans un contacte base de données, Journées Bases de données,AFCET-ADI,Dijon(France),113-138 1985.

HELD, G.D. ; INGRES a relational database systemp, Proc.AFIPS NCC, 44 : 409-416, 1975.

MEIER, A. ; A graph grammar approch to geographic data base, ETH report (Zurich), 49, 1982.

STONEBRAKER, M. ; The design and implementation of INGRES, Trans.on Database Systems ACM, 1(3) : 189-222, 1976.

THALMANN, D. ; MAGNENAT, N. Informatique graphique : conception et technique avec le langage MIRA, Gaëtan Morin Editeur, Québec, 1983.

YAMAGUCHI, K. ; KUNII, T. ELF : Extented relational model for large, flexible picture databases, Proc.of the IEEE WPDDM*,95-100, 1980.

YANG, G.Y.; A logical data organisation for the integrated databse of pictures and alphanumeric data, Proc.of the IEEE WPDDM* 158-164, 1980

ZLOOF, M.M.; Query by example, Proc.AFIPS NCC, 44 : 431-438, 1975.

ZLOOF, M.M. ; Query by example : a data base language, IBM System Journal, 16(4), 324-343, 1977.

---

* WPDD : "Workshop in picture data description and management" August 27-28 1980, A Silomar, Californie (USA)